AEROSPACE REPORT NO.
ATR-76(7361)-1, VOL V
(Formerly ATR-74(7341)-1)

DRA

Manned Systems Utilization Analysis (Study 2.1) Final Report

Volume V: Program Listing for the LOVES Computer Code

Prepared by STANLEY T. WRAY, JR.
Information Processing Division

1 September 1975

Prepared for OFFICE OF MANNED SPACE FLIGHT
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C.

Contract No. NASW-2727

Systems Engineering Operations
THE AEROSPACE CORPORATION

Report No.
ATR-76(7361)-1, Vol V
(Formerly ATR-74(7341)-7)

MANNED SYSTEMS UTILIZATION ANALYSIS (STUDY 2.1)

FINAL REPORT

Volume V: Program Listing for the LOVES Computer Code

Prepared by

Stanley T. Wray, Jr.
Data Processing Subdivision
Information Processing Division
Engineering Science Operations

1 September 1975

Systems Engineering Operations
THE AEROSPACE CORPORATION
El Segundo, California

Prepared for

OFFICE OF MANNED SPACE FLIGHT
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D. C.

Contract No. NASW-2727

Report No.
ATR-76(7361)-1, Vol V
(Formerly ATR-74(7341)-7)

MANNED SYSTEMS UTILIZATION ANALYSIS (STUDY 2.1)
FINAL REPORT
Volume V: Program Listing for the LOVES Computer Code

Prepared by

Stanley T. Wray, Jr.
Staff Engineer
Evaluation and Planning Systems
Data Processing Subdivision

Approved by

L. Sashkin, Director
Data Processing Subdivision
Information Processing Division
Engineering Science Operations

R. R. Wolfe
NASA Study 2.1 Director
Advanced Mission Analysis
Directorate
Advanced Orbital Systems Division

FOREWORD

The LOVES computer code was developed to investigate the concept of space servicing operational satellites as an alternative to replacing expendable satellites or returning satellites to earth for ground refurbishment. In addition to having the capability to simulate the expendable satellite operation and the ground refurbished satellite operation, the program is designed to simulate the logistics of space servicing satellites using an upper stage vehicle and/or the earth to orbit shuttle. The program not only provides for the initial deployment of the satellite but also simulates the random failure and subsequent replacement of various equipment modules comprising the satellite. The program has been used primarily to conduct trade studies and/or parametric studies of various space program operational philosophies.

The program was developed in the CDC 6400/7600 computer complex at The Aerospace Corporation, El Segundo, California, for implementation on a UNIVAC 1108 computer. It is coded in SIMSCRIPT 1.5 and FORTRAN IV. SIMSCRIPT (simulation of a program used for design and development purposes) is a simulation language originally developed at the Rand Corporation and now available from Consolidated Analysis Centers, Inc., (C.A.C.I.) in Santa Monica, California. FORTRAN IV (Formula Translation System) is a standard scientific programming language in common use in computer programs.

There are five volumes to this final report which are as follows:

Volume I:	Executive Summary, ATR-76(7361)-1, Vol I
Volume II:	Manned Systems Utilization, ATR-76(7361)-1, Vol II
Volume III:	LOVES Computer Simulations, Results and Analyses, ATR-76(7361)-1, Vol III
Volume IV:	Program Manual and Users Guide for the LOVES Computer Code, ATR-76(7361)-1, Vol IV (formerly ATR-74 (7341)-6)
Volume V:	Program Listing for the LOVES Computer Code, ATR-76(7361)-1, Vol V (formerly ATR-74(7341)-7)

This volume (Vol V) represents the final version of the program code. It incorporates all of the changes made to the code since the publication of the previous listing.

Design of the program was initiated by The Aerospace Corporation in FY 74 under Study 2.1, Operations Analysis, Payload Designs for Space Servicing (contract NASW 2575). It was completed in FY 75 under Study 2. Manned Systems Utilization Analysis (contract NASW 2727). The NASA Study Director for FY 74 and part of FY 75 was Mr. V. N. Huff, NASA Headquarters, Code MTE. The NASA Study Director for the balance of FY 75 was Dr. J. W. Steincamp, MSFC, Code PD 34.

```
 1UP      0      IC                                    DEFINE      2
 2DOWN    0      IC                                    DEFINE      3
 3OUT     0      IC                                    DEFINE      4
 4SHUT    0      IC                                    DEFINE      5
 5TUG     0      IC                                    DEFINE      6
 6SEPS    0      IC                                    DEFINE      7
 7BLANK   0      IC                                    DEFINE      8
 8G       0      FF                                    DEFINE      9
 9TIME3   0      FF                                    DEFINE     10
10TIMES   0      FF                                    DEFINE     11
11TIMEC   0      FI                                    DEFINE     12
12PDOWN   0      IC                                    DEFINE     13
13NINSU   0      IC                                    DEFINE     14
14WTSU    0      FC                                    DEFINE     15
15LENSU   0      FI                                    DEFINE     16
16NMD     0      IF                                    DEFINE     17
17SU      0      IF                                    DEFINE     18
18WAIT1   0     SF                                     DEFINE     19
19WAIT2   0     SF                                     DEFINE     20
20WSATU   0      FF                                    DEFINE     21
21WSATN   0      FF                                    DEFINE     22
22WMODU   0      FF                                    DEFINE     23
23WMODN   0      FF                                    DEFINE     24
24TRIG    0      II                                    DEFINE     25
25TRIG2   0      II                                    DEFINE     26
26TRIGS   0      II                                    DEFINE     27
27EXVEH   0      IF                                    DEFINE     28
28PREFT   0      FF                                    DEFINE     29
29TREFT   0      FF                                    DEFINE     30
30SREFT   0      FF                                    DEFINE     31
31PADT    0      FF                                    DEFINE     32
32DAYS    0      FF                                    DEFINE     33
33DV      0      FF                                    DEFINE     34
34FISP    0      FF                                    DEFINE     35
35WD      0      FF                                    DEFINE     36
36WPNU    0      FF                                    DEFINE     37
37WCONS   0      FF                                    DEFINE     38
38IORB    0      FI                                    DEFINE     39
39NQ      0     SI                                     DEFINE     40
40RA      0      FF                                    DEFINE     41
41VCO     0      FF                                    DEFINE     42
42RO      0      FF                                    DEFINE     43
43P1      0      FF                                    DEFINE     44
44RTFLG   0      FI                                    DEFINE     45
45PALEN   0      IF                                    DEFINE     46
46FLYT    0      FF                                    DEFINE     47
47WAIT3   0     SF                                     DEFINE     48
48TLIMS   0      FF                                    DEFINE     49
49TRIN    0      FF                                    DEFINE     50
                                                       DEFINE     51
                                                       DEFINE     52
+N START4        N TIME1  3      F                     DEFINE     53
                 N PSAT   31/2   I                     DEFINE     54
                 N TIMEV  2      F                     DEFINE     55
                 N PMOD   32/2   I                     DEFINE     56
+N TERM 4        N TIME2  4      F                     DEFINE     57
+N NWSAT4        N VNAME  4      F                     DEFINE     58
                 N TIMEA  4      F
```

```
                N LQEV  3       I                                                    DEFINE     59
N ARRIV4                                                                             DEFINE     60
N BACK 4                                                                             DEFINE     61
N FAIL 4                                                                             DEFINE     62
N LAUNC4                                                                             DEFINE     63
N NEWME4                                                                             DEFINE     64
N REFMO4                                                                             DEFINE     65
N REFSA4                                                                             DEFINE     66
N REFVE4                                                                             DEFINE     67
N REMOV4                                                                             DEFINE     68
N RETRI4                                                                             DEFINE     69
N SATON4                                                                             DEFINE     70
N WARN 4                                                                             DEFINE     71
N QWAIT4                                                                             DEFINE     72
                                                                                     DEFINE     73
T NEW  2        T SNEWS  11/2  I 50FNEWS  0     I    NEWS  L                       DEFINE     74
                T SCHSY  12/2  I                                                     DEFINE     75
                T SCHDT  2     F                                                     DEFINE     76
T FR   4        T SFRS   11/2  I 51FFRS   0     I    FRS   XSATNO L                DEFINE     77
                T PFRS   12/2  I 52LFRS   0     I                                  DEFINE     78
                T TIMEF  2     F                                                     DEFINE     79
                T SATNO  31/4  I                                                     DEFINE     80
                T ST     32/4  I                                                     DEFINE     81
                T SATSY  33/4  I                                                     DEFINE     82
                T NPS    34/4  I                                                     DEFINE     83
                T MODNO  41/2  I                                                     DEFINE     84
                T NOSTA  43/4  I                                                     DEFINE     85
                T NDEL   44/4  I                                                     DEFINE     86
                T W1     1     F                                                     DEFINE     87
                T W2     2     F                                                     DEFINE     88
                T W3     3     F                                                     DEFINE     89
                T W4     4     F                                                     DEFINE     90
T MESET4        T SMES   11/2  I 53FMES  0     I    MES   L                       DEFINE     91
                T MEDT   2     F                                                     DEFINE     92
                         54IQ     0     I                                          DEFINE     93
                         55SEPFT  0     F                                          DEFINE     94
                         56MODB   0     I                                          DEFINE     95
                         57EXMOD  0     I                                          DEFINE     96
                         58DELTA  0     F                                          DEFINE     97
                         59EXTUG  0     F                                          DEFINE     98
                                                                                   DEFINE     99
                         60NORBS  0     IC                                         DEFINE    100
                         61ORBID  1     I                                          DEFINE    101
                         62ORBDV  1     F                                          DEFINE    102
                         63ORBPD  1     F                                          DEFINE    103
                         64ORBRA  1     F                                          DEFINE    104
                         65ORBVC  1     F                                          DEFINE    105
                         66ORBTM  1     F                                          DEFINE    106
                         67RQUP   1     I                                          DEFINE    107
                         68RQSEP  1     I                                          DEFINE    108
                         69RQSUT  1     I                                          DEFINE    109
                         70PQUE   1     I                                          DEFINE    110
                         71NL     1    SI                                          DEFINE    111
                         72ANMD   1     F                                          DEFINE    112
                         73W      1    SF                                          DEFINE    113
                         74NMDFL  1     I                                          DEFINE    114
T PAYLD3        T SORBQ  11/2  I 75FORBQ  1     I    ORBQ1 XLQTIM L               DEFINE    115
```

```
I  CORBQ  12/2  I  76LORBQ 1     I
I  ISAT   23/4  I
I  IMOD   21/2  I
I  IRT    24/4  I
I  ANGLE  3     SF
I  PAYWT  4     F
I  PAYLN  5     F
I  GOTIM  6     F
I  LQTIM  7     F
I  CITEM  82/2  I
I  MLEV   81/2  I
                77OV1    1     F
                78EXOR3  1     FI
                79MODS   0     I

                80IL     0     IC
                81FLTIM  1     F
                82ILOAD  1     I
                83PANGL  1     SF
                84TIMEG  0     FC

                85NVEH   0     IC
                86NAMEV  1     I
                87DAYSV  1     F
                88ISPV   1     F
                89WDV    1     F
                90WPNUV  1     F
                91WCONV  1     F
                92REFTV  1     F
                93EXPV   1     F
                94PAYLV  1     F
                95IDV    1     I
                96NSTAG  1     I
                97SQLID  1     I
                98PSERV  0     IC
                99WAIT4  0     F

                100NYEAR 0     IC
                101SUTFY 1     I
                102SUM90 1     I
                103MAX90 1     I
                104MIN90 1     I
                105TUGFY 1     I
                106SUM39 1     I
                107MAX39 1     I
                108MIN39 1     I
                109SEPFY 1     I
                110SUM86 1     I
                111MAX86 1     I
                112MIN86 1     I
                113LIMIT 0     F

                114NPAD  0     IC
                115VPAD  1     I
                116IPAD  0     I
                117NOTUG 0     IC
                118TUP   0     F
```

```
DEFINE  116
DEFINE  117
DEFINE  118
DEFINE  119
DEFINE  120
DEFINE  121
DEFINE  122
DEFINE  123
DEFINE  124
DEFINE  125
DEFINE  126
DEFINE  127
DEFINE  128
DEFINE  129
DEFINE  130
DEFINE  131
DEFINE  132
DEFINE  133
DEFINE  134
DEFINE  135
DEFINE  136
DEFINE  137
DEFINE  138
DEFINE  139
DEFINE  140
DEFINE  141
DEFINE  142
DEFINE  143
DEFINE  144
DEFINE  145
DEFINE  146
DEFINE  147
DEFINE  148
DEFINE  149
DEFINE  150
DEFINE  151
DEFINE  152
DEFINE  153
DEFINE  154
DEFINE  155
DEFINE  156
DEFINE  157
DEFINE  158
DEFINE  159
DEFINE  160
DEFINE  161
DEFINE  162
DEFINE  163
DEFINE  164
DEFINE  165
DEFINE  166
DEFINE  167
DEFINE  168
DEFINE  169
DEFINE  170
DEFINE  171
DEFINE  172
```

```
+                    119TDOWN  0     F                              DEFINE    173
+                                                                   DEFINE    174
+                    120NSHUT  0     IC                             DEFINE    175
+                    121VSHUT  1     I                              DEFINE    176
+                    122ISHUT  0     I                              DEFINE    177
+                    123MFSUT  0     I                              DEFINE    178
+                    124IFSUT  0     I                              DEFINE    179
+                    125NFSUT  0     I                              DEFINE    180
+                    126WUSEP  0     F                              DEFINE    181
+                    127WDNSP  0     F                              DEFINE    182
+                    128LSEP   0     F                              DEFINE    183
+                    129QUIT   0     I                              DEFINE    184
+                                                                   DEFINE    185
+                    130NTUG   0     IC                             DEFINE    186
+                    131VTUG   1     I                              DEFINE    187
+                    132ITUG   0     I                              DEFINE    188
+                    133NTFLT  0     I                              DEFINE    189
+                    134ITFLT  0     I                              DEFINE    190
+                    135MTFLT  0     I                              DEFINE    191
+                                                                   DEFINE    192
+                    140NSEPS  0     IC                             DEFINE    193
+                    141VSEPS  1     I                              DEFINE    194
+                    142ISEPS  0     I                              DEFINE    195
+                    143NFSEP  0     I                              DEFINE    196
+                    144IFSEP  0     I                              DEFINE    197
+                    145MFSEP  0     I                              DEFINE    198
+                    146AVSEP  1     F                              DEFINE    199
+                    147SWON   1     F                              DEFINE    200
+                    148SLDN   1     F                              DEFINE    201
+                                                                   DEFINE    202
+                    150MITAB  0     IC                             DEFINE    203
+                    151MNAME  1     I                              DEFINE    204
+                    152ALPF   1     F                              DEFINE    205
+                    1538ETAF  1     F                              DEFINE    206
+                    154TTFMD  1     F                              DEFINE    207
+                    155ALPW   1     F                              DEFINE    208
+                    156BETAW  1     F                              DEFINE    209
+                    157TTWMD  1     F                              DEFINE    210
+                    158MODWT  1     F                              DEFINE    211
+                    159MDVOL  1     F                              DEFINE    212
+                    160MCLAS  1     I                              DEFINE    213
+                    161MDCNT  1     I                              DEFINE    214
+                    162S121   1     I                              DEFINE    215
+                    163X121   1     I                              DEFINE    216
+                    164N121   1     I                              DEFINE    217
+                    165NOWAR  1     I                              DEFINE    218
+                    166S125   1     I                              DEFINE    219
+                    167X125   1     I                              DEFINE    220
+                    168N125   1     I                              DEFINE    221
+                    169NOFAL  1     I                              DEFINE    222
+                    170S129   1     I                              DEFINE    223
+                    171X129   1     I                              DEFINE    224
+                    172N129   1     I                              DEFINE    225
+                                                                   DEFINE    226
+                    179CHEM   0     I                              DEFINE    227
+                    180SITAB  0     IC                             DEFINE    228
+                    181SNAME  1     I                              DEFINE    229
```

```
                                182SWT   1     F                                DEFINE     230
                                183SVOL  1     F                                DEFINE     231
                                184PRIOR 1     F                                DEFINE     232
                                185INCL  1     F                                DEFINE     233
                                186ORBIT 1     I                                DEFINE     234
                                187TTSAT 1     F                                DEFINE     235
                                188PTSAT 1     F                                DEFINE     236
                                188EXWT  1     F                                DEFINE     237
                                189EXSAT 1     I                                DEFINE     238
                                190NRSAT 1     I                                DEFINE     239
                                191NMODS 1     I                                DEFINE     240
                                192POLDN 1     I                                DEFINE     241
                                193SORTE 1     F                                DEFINE     242
T MDSAT2        T SMDS  11/2 I  194FMDS  1     I     MDS 1 F                    DEFINE     243
                                195LMDS  1     I                                DEFINE     244
                                                                                DEFINE     245
                                197NEXIT 1     I                                DEFINE     246
                                198LEXIT 1     I                                DEFINE     247
                                199MSEP  1     I                                DEFINE     248
                                200STST3 0     IC                               DEFINE     249
                                201SYNAM 1     I                                DEFINE     250
                                202TTSYS 1     F                                DEFINE     251
                                203PTTSY 1     F                                DEFINE     252
                                204NSAT  1     I                                DEFINE     253
                                205FSAT  1     I                                DEFINE     254
                                206LSAT  1     I                                DEFINE     255
                                207STAT  1     I                                DEFINE     256
                                208NFUP  1     I                                DEFINE     257
                                209TGOSY 1     F                                DEFINE     258
                                210SYLF  1     F                                DEFINE     259
                                211XSYLF 1     F                                DEFINE     260
                                212NSYLF 1     F                                DEFINE     261
                                213BEGSY 1     F                                DEFINE     262
                                214HALSY 1     F                                DEFINE     263
                                215TLASY 1     SF                               DEFINE     264
                                216SDTSY 1     F                                DEFINE     265
                                217PERSY 1     F                                DEFINE     266
                                218X200  1     F                                DEFINE     267
                                219N200  1     F                                DEFINE     268
                                220DNTSY 1     F                                DEFINE     269
                                221C208  1     F                                DEFINE     270
                                222X208  1     F                                DEFINE     271
                                223N208  1     F                                DEFINE     272
                                                                                DEFINE     273
                                230SYORB 0     IC                               DEFINE     274
                                231ITSAT 1     I                                DEFINE     275
                                232ITSYS 1     I                                DEFINE     276
                                233SSTAT 1     I                                DEFINE     277
                                234PHASE 1     SF                               DEFINE     278
                                235ATIME 1     F                                DEFINE     279
                                236DTIME 1     F                                DEFINE     280
                                237MARKS 1     I                                DEFINE     281
                                238MARKU 1     I                                DEFINE     282
                                239MARKD 1     I                                DEFINE     283
                                240LFSAT 1     F                                DEFINE     284
                                241SUMSL 1     F                                DEFINE     285
                                242MAXSL 1     F                                DEFINE     286
```

```
                                          243MINSL  1     F                          DEFINE   287
                                          244TGO    1     F                          DEFINE   288
                                          245BEGST  1     F                          DEFINE   289
                                          246HALST  1     F                          DEFINE   290
                                          247TLAST  1    SF                          DEFINE   291
                                          248SDTST  1     F                          DEFINE   292
                                          249PERST  1     F                          DEFINE   293
                                          250X216   1     F                          DEFINE   294
                                          251N216   1     F                          DEFINE   295
                                          252DNTST  1     F                          DEFINE   296
                                          253C223   1     F                          DEFINE   297
                                          254X223   1     F                          DEFINE   298
                                          255N223   1     F                          DEFINE   299
                                          256SATLF  1     F                          DEFINE   300
                                          257S227   1     F                          DEFINE   301
                                          258X227   1     F                          DEFINE   302
                                          259N227   1     F                          DEFINE   303
                                          260NPOS   1     I                          DEFINE   304
                                          261NDEP   1     I                          DEFINE   305
T MODSY8                T SMOD   11/2 I262FMOD  1     I    MOD 1 F               DEFINE   306
                        T NOMOD  12/2 I263LMOD  1     I                          DEFINE   307
                        T EFAIL  21/2 I                                          DEFINE   308
                        T NUM    23/4 I                                          DEFINE   309
                        T NRU    24/4 I                                          DEFINE   310
                        T MAXNU  31/4 I                                          DEFINE   311
                        T MINNU  32/4 I                                          DEFINE   312
                        T MSTAT  33/4 I                                          DEFINE   313
                        T SUMNU  34/4 I                                          DEFINE   314
                        T LOADF  41/3 I                                          DEFINE   315
                        T SUMLF  42/3 I                                          DEFINE   316
                        T MAXLF  43/3 I                                          DEFINE   317
                        T MINLF  51/3 I                                          DEFINE   318
                        T EDO    52/3 I                                          DEFINE   319
                        T EWARN  61/2 I                                          DEFINE   320
                        T MNO    62/2 I                                          DEFINE   321
                                          264XSAT   1     I                          DEFINE   322
                                          270NVS    0     IC                         DEFINE   323
                                          271CVA    1     F                          DEFINE   324
                                          272TCVA   1     F                          DEFINE   325
                                          273XCVA   1     F                          DEFINE   326
                                          274MCVA   1     F                          DEFINE   327
                                          275VDATE  1     F                          DEFINE   328
                                          276VTD    1     F                          DEFINE   329
                                          277XTD    1     F                          DEFINE   330
                                          278MTD    1     F                          DEFINE   331
                                                                                     DEFINE   332
                                          280CTUG   1     F                          DEFINE   333
                                          281WTUG   1     F                          DEFINE   334
                                          282CSHUT  1     F                          DEFINE   335
                                          283WSHUT  1     F                          DEFINE   336
                                          284CSEPS  1     F                          DEFINE   337
                                          285WSEPS  1     F                          DEFINE   338
                                          286NPAD1  1     I                          DEFINE   339
                                          287NPAD2  1     I                          DEFINE   340
                                          288CDTUG  1     F                          DEFINE   341
                                          289WDTUG  1     F                          DEFINE   342
                                          290CDSUT  1     F                          DEFINE   343
```

```
+                          291WDSUI  1     F                          DEFINE    344
+                          292CDSEP  1     F                          DEFINE    345
+                          293WDSEP  1     F                          DEFINE    346
+                          297FREE   0     I                          DEFINE    347
+                          298SCOOT  0     I                          DEFINE    348
+                          299IEVQW  0     I                          DEFINE    349
+                          300IEVST  0     I                          DEFINE    350
+                          301IEVTE  0     I                          DEFINE    351
+                          302IEVNW  0     I                          DEFINE    352
+                          303IEVWA  0     I                          DEFINE    353
+                          304IEVFA  0     I                          DEFINE    354
+                          305IEVLA  0     I                          DEFINE    355
+                          306IEVAR  0     I                          DEFINE    356
+                          307IEVVE  0     I                          DEFINE    357
+                          308IEVMO  0     I                          DEFINE    358
+                          309IEVBE  0     I                          DEFINE    359
+                          310IEVSA  0     I                          DEFINE    360
+                          311IEVOV  0     I                          DEFINE    361
+                          312IEVRI  0     I                          DEFINE    362
+                          313IEVDN  0     I                          DEFINE    363
+                          314IEVME  0     I                          DEFINE    364
+                          315NSUUP  0     I                          DEFINE    365
+                          316NSUDN  0     I                          DEFINE    366
+                          317MODOF  0     I                          DEFINE    367
+                          318SATOF  0     I                          DEFINE    368
+                          319MIXED  0     I                          DEFINE    369
+                          320FWAIT  0     I                          DEFINE    370
+                          321FWGT   0     I                          DEFINE    371
      EVENTS                                                          EVENTS      2
      1 EXOGENOUS                                                     EVENTS      3
          BEGIN (1)                                                   EVENTS      4
      16 ENDOGENOUS                                                   EVENTS      5
          QWAIT                                                       EVENTS      6
          START                                                       EVENTS      7
          TERM                                                        EVENTS      8
          NWSAT                                                       EVENTS      9
          WARN                                                        EVENTS     10
          FAIL                                                        EVENTS     11
          LAUNC                                                       EVENTS     12
          ARRIV                                                       EVENTS     13
          REFVE                                                       EVENTS     14
          REFMO                                                       EVENTS     15
          REFSA                                                       EVENTS     16
          REMOV                                                       EVENTS     17
          RETRI                                                       EVENTS     18
          SATDN                                                       EVENTS     19
          NEWME                                                       EVENTS     20
          BACK                                                        EVENTS     21
       END                                                            EVENTS     22
      SUBROUTINE ADMOD(IS,IM)                                         ADMOD      2
C                                                                     ADMOD      3
C     THIS ROUTINE CREATES THE FAILURE AND WARNING OF A MODULE        ADMOD      4
C                                                                     ADMOD      5
      LET EOO(IM) = 0                                                 ADMOD      6
      LET MSTAT(IM) =    2                                            ADMOD      7
      IF TIME GT TIMES, RETURN                                        ADMOD      8
      LET I = NOMOD(IM)                                               ADMOD      9
```

```
      CALL WEIBUL(ALPW(I),BETAW(I),TW,ALPF(I),BETAF(I),TF)                ADMOD   10
C                                                                         ADMOD   11
CC    CAUSE WARNINGS                                                      ADMOD   12
C                                                                         ADMOD   13
      LET IEW = EWARN(IM)                                                 ADMOD   14
      IF IEW EQ 0, GO TO 2                                                ADMOD   15
      IF TIMEF(IEW) NE 0., CANCEL WARN CALLED IEW                         ADMOD   16
      DESTROY WARN CALLED IEW                                             ADMOD   17
      LET EWARN(IM) = 0                                                   ADMOD   18
    2 IF TW EQ 0., GO TO 5                                                ADMOD   19
      LET TX = TTFMD(I)-WMODU                                             ADMOD   20
      IF TW GT TX, LET TW = TX                                            ADMOD   21
      IF TIME + TW GT TGO(IS), GO TO 5                                    ADMOD   22
      CREATE WARN CALLED IEW                                              ADMOD   23
      LET PSAT(IEW) = IS                                                  ADMOD   24
      LET PMOD(IEW) = IM                                                  ADMOD   25
      LET TIMEA(IEW) = ATIME(IS)                                          ADMOD   26
      CAUSE WARN CALLED IEW AT TIME + TW                                  ADMOD   27
C                                                                         ADMOD   28
CC    CAUSE FAILURES                                                      ADMOD   29
C                                                                         ADMOD   30
    5 LET EWARN(IM) = IEW                                                 ADMOD   31
      LET IEF = EFAIL(IM)                                                 ADMOD   32
      IF TF GT TTFMD(I), LET TF = TTFMD(I)                                ADMOD   33
      IF IEF EQ 0, GO TO 6                                                ADMOD   34
      IF TIMEF(IEF) NE 0., CANCEL FAIL CALLED IEF                         ADMOD   35
      DESTROY FAIL CALLED IEF                                             ADMOD   36
      LET EFAIL(IM) = 0                                                   ADMOD   37
    6 IF TF EQ 0., GO TO 10                                               ADMOD   38
      IF TIME + TF GT TGO(IS), GO TO 10                                   ADMOD   39
      CREATE FAIL CALLED IEF                                              ADMOD   40
      LET PSAT(IEF) = IS                                                  ADMOD   41
      LET PMOD(IEF) = IM                                                  ADMOD   42
      LET TIMEA(IEF) = ATIME(IS)                                          ADMOD   43
      CAUSE FAIL CALLED IEF AT TIME + TF                                  ADMOD   44
      LET EFAIL(IM) = IEF                                                 ADMOD   45
   10 RETURN                                                              ADMOD   46
      END                                                                 ADMOD   47
      ENDOGENOUS EVENT ARRIV                                              ARRIV    2
C                                                                         ARRIV    3
CC    THIS IS THE ARRIVAL OF A SATELLITE IN ORBIT AFTER TIME OF FLIGHT.   ARRIV    4
C                                                                         ARRIV    5
CC    NOW ACTIVATE NEW SATELLITES                                         ARRIV    6
C                                                                         ARRIV    7
CC         ATTEMPT TO REACTIVATE SATELLITES WITH REPLACED MODULES         ARRIV    8
C                                                                         ARRIV    9
      LET IEVAR = IEVAR + 1                                               ARRIV   10
      LET IS = PSAT(ARRIV)                                                ARRIV   11
      LET IM = PMOD(ARRIV)                                                ARRIV   12
      DESTROY ARRIV                                                       ARRIV   13
      IF IM NE 0, GO TO 100                                               ARRIV   14
      LET JST = ITSAT(IS)                                                 ARRIV   15
      LET JSY = ITSYS(IS)                                                 ARRIV   16
      LET NDEP(IS) = NDEP(IS) + 1                                         ARRIV   17
      LET NPOS(IS) = NPOS(IS) + 1                                         ARRIV   18
      LET K = 0                                                           ARRIV   19
      DO TO 2, FOR I=(FSAT(JSY))(LSAT(JSY))                               ARRIV   20
```

```
      IF NPOS(I) NE 0, LET K = K + 1                                    ARRIV   21
2     LOOP                                                              ARRIV   22
      IF K GE NFUP(JSY), LET TGOSY(JSY) = TIME + TTSYS(JSY)             ARRIV   23
      IF BEGST(IS) EQ 0., LET BEGST(IS) = TIME                          ARRIV   24
      IF TLAST(IS) EQ 0., LET TLAST(IS) = -TIME                         ARRIV   25
      LET ATIME(IS) = TIME                                              ARRIV   26
      LET DTIME(IS) = TIME                                              ARRIV   27
      LET TGO(IS) = TIME + TTSAT(JST)                                   ARRIV   28
      IF TGO(IS) GT TIMES, LET TGO(IS) = TIMES                          ARRIV   29
      IF TGOSY(JSY) EQ 0., GO TO 5                                      ARRIV   30
      IF TGOSY(JSY) GT TIMES, LET TGOSY(JSY)= TIMES                     ARRIV   31
      DO TO 4, FOR I = (FSAT(JSY))(LSAT(JSY))                           ARRIV   32
      IF TGO(I) LE TGOSY(JSY), GO TO 4                                  ARRIV   33
C                                                                       ARRIV   34
CC    RESCHEDULE SATELLITE TERMINATIONS CAREFULLY                       ARRIV   35
CC    ------------------------------------------                        ARRIV   36
C                                                                       ARRIV   37
      LET T =TGOSY(JSY)                                                 ARRIV   38
      IF MARKS(I) EQ 0, GO TO 20                                        ARRIV   39
      CANCEL SATDN CALLED MARKS(I)                                      ARRIV   40
      CAUSE SATDN CALLED MARKS(I) AT T                                  ARRIV   41
20    IF MARKU(I) EQ 0, GO TO 30                                        ARRIV   42
      CANCEL NWSAT CALLED MARKU(I)                                      ARRIV   43
      DESTROY NWSAT CALLED MARKU(I)                                     ARRIV   44
      LET MARKU(I) = 0                                                  ARRIV   45
30    DO TO 40, FOR ALL MODSY IN MOD(I)                                 ARRIV   46
      IF EWARN(MODSY) EQ 0, GO TO 35                                    ARRIV   47
      IF TIMEV(EWARN(MODSY)) LE T, GO TO 35                             ARRIV   48
      CANCEL WARN CALLED EWARN(MODSY)                                   ARRIV   49
      DESTROY WARN CALLED EWARN(MODSY)                                  ARRIV   50
      LET EWARN(MODSY) = 0                                              ARRIV   51
35    IF EFAIL(MODSY) EQ 0, GO TO 40                                    ARRIV   52
      IF TIMEV(EFAIL(MODSY)) LE T, GO TO 40                             ARRIV   53
      CANCEL FAIL CALLED EFAIL(MODSY)                                   ARRIV   54
      DESTROY FAIL CALLED EFAIL(MODSY)                                  ARRIV   55
      LET EFAIL(MODSY) = 0                                              ARRIV   56
40    LOOP                                                              ARRIV   57
    3 LET TGO(I) = TGOSY(JSY)                                           ARRIV   58
4     LOOP                                                              ARRIV   59
      IF BEGSY(JSY) EQ 0., LET BEGSY(JSY) = TIME                        ARRIV   60
      IF TLASY(JSY) EQ 0., LET TLASY(JSY) = -TIME                       ARRIV   61
5     CALL STATUS(IS,0,2)                                               ARRIV   62
      CALL ADMOD(IS,MODSY), FOR ALL MODSY IN MOD( . .IS)                ARRIV   63
      LET IPOL = POLDN(JST)                                             ARRIV   64
      IF IPOL EQ 0, GO TO 200                                           ARRIV   65
      LET T = TIME + TTSAT(JST) + WAIT1                                 ARRIV   66
      CALL SAVER(T,IS)                                                  ARRIV   67
C                                                                       ARRIV   68
CC    SCHEDULE SATELLITE EVENT (SATDN) AT TERMINATION TIME              ARRIV   69
C                                                                       ARRIV   70
  200 IF MARKS(IS) EQ 0, GO TO 1                                        ARRIV   71
      CANCEL SATDN CALLED MARKS(IS)                                     ARRIV   72
      DESTROY SATDN CALLED MARKS(IS)                                    ARRIV   73
      LET MARKS(IS) = 0                                                 ARRIV   74
    1 LET T = TIME + TTSAT(JST)                                         ARRIV   75
      IF SORTE(ITSAT(IS)) NE 0., RETURN                                 ARRIV   76
      IF T GT TGO(IS), LET T = TGO(IS)                                  ARRIV   77
```

```
      IF I.LT.TIME, RETURN                                              ARRIV    78
      CREATE SATON CALLED MARKS(IS)                                     ARRIV    79
      LET PSAT(MARKS(IS)) = IS                                          ARRIV    80
      CAUSE SATON CALLED MARKS(IS) AT T                                 ARRIV    81
      RETURN                                                            ARRIV    82
C                                                                       ARRIV    83
CC    SINGLE MODULE IS REPLACED IN ORBIT                                ARRIV    84
C                                                                       ARRIV    85
  100 IF SSTAT(IS) EQ OUT, RETURN                                       ARRIV    86
      CALL ADMOD(IS,IM)                                                 ARRIV    87
      CALL STATUS(IS,IM,2)                                              ARRIV    88
      LET MDCNT(NOMOD(IM)) = MDCNT(NOMOD(IM)) + 1                       ARRIV    89
      RETURN                                                            ARRIV    90
      END                                                               ARRIV    91
      ENDOGENOUS EVENT BACK                                             BACK      2
C                                                                       BACK      3
CCC   WHEN THIS EVENT OCCURS, THE SATELLITE IS REMOVED FROM ORBIT       BACK      4
C                                                                       BACK      5
      CALL STATUS(PSAT(BACK),0,6)                                       BACK      6
      DESTROY BACK                                                      BACK      7
      RETURN                                                            BACK      8
      END                                                               BACK      9
      EXOGENOUS EVENT BEGIN                                             BEGIN     2
      SAVE                                                              BEGIN     3
      READ TIMEB,TIMES                                                  BEGIN     4
      FORMAT(2M5.2.2)                                                   BEGIN     5
      CREATE START                                                      BEGIN     6
      CAUSE START AT 1.                                                 BEGIN     7
      CALL LDAT                                                         BEGIN     8
      LET IEVBE = IEVBE + 1                                             BEGIN     9
C                                                                       BEGIN    10
CCC   INITIALIZATION                                                    BEGIN    11
C                                                                       BEGIN    12
      LET TREFT = TREFT/360.                                            BEGIN    13
      LET SREFT = SREFT/360.                                            BEGIN    14
      LET PREFT = PREFT/360.                                            BEGIN    15
      LET SEPFT =SEPFT/360.                                             BEGIN    16
      LET WAIT3 = WAIT3/360.                                            BEGIN    17
      LET PADT  = PADT /360.                                            BEGIN    18
      LET WAIT1 = WAIT1/360.                                            BEGIN    19
      LET WAIT2 = WAIT2/360.                                            BEGIN    20
      LET WAIT4 = WAIT4/360.                                            BEGIN    21
      LET WSATU = WSATU/360.                                            BEGIN    22
      LET WSATN = WSATN/360.                                            BEGIN    23
      LET WMODU = WMODU/360.                                            BEGIN    24
      LET WMODN = WMODN/360.                                            BEGIN    25
      LET NTFLT = 1000                                                  BEGIN    26
      LET TLIMS = TLIMS/360.                                            BEGIN    27
      LET NFSEP = 1000                                                  BEGIN    28
      LET NFSUT = 1000                                                  BEGIN    29
      LET MIN39(I)  = 1000, FOR I=(1)(NYEAR)                            BEGIN    30
      LET MIN86(I)  = 1000, FOR I=(1)(NYEAR)                            BEGIN    31
      LET MIN90(I)  = 1000, FOR I=(1)(NYEAR)                            BEGIN    32
      LET MINSL(I) =1000, FOR I=(1)(SYORB)                              BEGIN    33
      LET N227(I)  = 1000, FOR I=(1)(SYORB)                             BEGIN    34
      LET N208(I)  = 1000., FOR I=(1)(STSTB)                            BEGIN    35
      LET N200(I)  = 1000., FOR I=(1)(STSTB)                            BEGIN    36
```

```
      LET N223(I) = 1000.; FOR I=(1)(SYORB)                          BEGIN    37
      LET N216(I) = 1000.; FOR I=(1)(SYORB)                          BEGIN    38
      LET N121(I) = 1000., FOR I=(1)(MITAB)                          BEGIN    39
      LET N125(I) = 1000, FOR I=(1)(MITAB)                           BEGIN    40
      LET N129(I) = 1000, FOR I=(1)(MITAB)                           BEGIN    41
      LET MTD(I) = 1000.,FOR I=(1)(3)                                BEGIN    42
      LET MCVA(I) = 1000., FOR I=(1)(3)                              BEGIN    43
      RETURN                                                         BEGIN    44
      END                                                            BEGIN    45
      SUBROUTINE CSPAY                                               CSPAY    2
C                                                                    CSPAY    3
CC    COMPUTE LAUNCH STATISTICS FOR PAYLOADS                         CSPAY    4
C                                                                    CSPAY    5
      LET B = 0.                                                     CSPAY    6
      DO TO 11, FOR I=(1)(NL(IORB))                                  CSPAY    7
      LET NY = ILOAD(I)                                              CSPAY    8
      LET B = B + PAYWT(NY)                                          CSPAY    9
      IF IMOD(NY) EQ 0, GO TO 11                                     CSPAY   10
      LET NX = IMOD(NY)                                              CSPAY   11
      LET NUM(NX) = NUM(NX) + 1                                      CSPAY   12
   11 LOOP                                                           CSPAY   13
      LET NMD = ANMD(IORB)                                           CSPAY   14
      LET SU = (NMD+NINSU-1)/NINSU                                   CSPAY   15
      LET X = 0                                                      CSPAY   16
      IF SU EQ 0., GO TO 13                                          CSPAY   17
      LET X = SU*WTSU/ANMD(IORB)                                     CSPAY   18
      LET B = B + SU*WTSU                                            CSPAY   19
   13 DO TO 14, FOR J=(1)(NL(IORB))                                  CSPAY   20
      LET II = ILOAD(J)                                              CSPAY   21
      LET NX = ISAT(II)                                              CSPAY   22
      LET NY = IMOD(II)                                              CSPAY   23
      IF NY EQ 0, GO TO 12                                           CSPAY   24
      LET PAYWT(II) = PAYWT(II) + X                                  CSPAY   25
      LET M = 100.*PAYWT(II)/B + .5                                  CSPAY   26
      LET LOADF(NY) = LOADF(NY) + M                                  CSPAY   27
      GO TO 15                                                       CSPAY   28
   12 LET SATLF(NX) = SATLF(NX) + PAYWT(II)/B                        CSPAY   29
   15 LET LFSAT(NX) = LFSAT(NX) + PAYWT(II)/B                        CSPAY   30
   14 LOOP                                                           CSPAY   31
      RETURN                                                         CSPAY   32
      END                                                            CSPAY   33
      SUBROUTINE DROPQ(J,IO)                                         DROPQ    2
C                                                                    DROPQ    3
CC    DROP PAYLOAD J FROM LOAD QUEUE ORB(IO)                         DROPQ    4
C                                                                    DROPQ    5
      REMOVE J FROM ORBQ(IO)                                         DROPQ    6
      LET K = MLEV(J)                                                DROPQ    7
      DESTROY PAYLD CALLED J                                         DROPQ    8
      IF K.EQ.0, RETURN                                              DROPQ    9
      CANCEL LAUNC CALLED K                                          DROPQ   10
      DESTROY LAUNC CALLED K                                         DROPQ   11
      RETURN                                                         DROPQ   12
      END                                                            DROPQ   13
      ENDOGENOUS EVENT FAIL                                          FAIL     2
C                                                                    FAIL     3
CCC   THIS ROUTINE WILL MARK OUTAGE OF A SATELLITE AND NOTE WHICH MODULE FAIL     4
C                                                                    FAIL     5
```

```
C      IS OUT (MAYBE MORE THAN ONE).                                    FAIL      6
C                                                                       FAIL      7
       LET IEVFA = IEVFA + 1                                            FAIL      8
       IF TIME GE TIMEG, LET EXMOD = MODS                               FAIL      9
       LET IS = PSAT (FAIL)                                             FAIL     10
       LET IM = PMOD (FAIL)                                             FAIL     11
       LET T = TIMEA(FAIL)                                              FAIL     12
       DESTROY FAIL                                                     FAIL     13
       LET EFAIL(IM) = 0                                                FAIL     14
C                                                                       FAIL     15
CC     BLOCK FAILURE EVENT (FOR LAUNCH) IF MODULE IS NOT REPLACEABLE    FAIL     16
C                                                                       FAIL     17
       IF SSTAT(IS) EQ OUT, RETURN                                      FAIL     18
       IF T LT ATIME(IS), RETURN                                        FAIL     19
       CALL STATUS(IS,IM,3)                                             FAIL     20
       LET NOFAL(NOMOD(IM)) = NOFAL(NOMOD(IM)) + 1                      FAIL     21
       IF XSAT(IS) EQ 100, RETURN                                       FAIL     22
       IF SSTAT(IS) EQ OUT, RETURN                                      FAIL     23
       LET DELAY = WMODN                                                FAIL     24
C                                                                       FAIL     25
CC   BLOCK EVENT AFTER TIMES                                            FAIL     26
C                                                                       FAIL     27
       IF EWARN(IM) NE 0, RETURN                                        FAIL     28
       IF TIME + DELAY GT TGO(IS), RETURN                               FAIL     29
C                                                                       FAIL     30
CC     PUT FAILURES INTO FREEBIE QUEUE                                  FAIL     31
C                                                                       FAIL     32
       CREATE QWAIT                                                     FAIL     33
       LET PSAT(QWAIT) = IS                                             FAIL     34
       LET PMOD(QWAIT) = IM                                             FAIL     35
       LET TIMEA(QWAIT) = DELAY                                         FAIL     36
       CAUSE QWAIT AT TIME + WAIT4                                      FAIL     37
       RETURN                                                           FAIL     38
       END                                                              FAIL     39
       SUBROUTINE FILEO                                                 FILEO     2
C                                                                       FILEO     3
CC   OUTPUT SATELLITE STATUS SUMMARY REPORT                             FILEO     4
CC     CHRONOLOGICAL HISTORY OF EVENTS PRESENTED BY SATELLITE           FILEO     5
C                                                                       FILEO     6
       DIMENSION WWW(4)                                                 FILEO     7
       LET TRIG2 = 1                                                    FILEO     8
       WRITE ON 6                                                       FILEO     9
       FORMAT(*1*)                                                      FILEO    10
C                                                                       FILEO    11
CC     READ DATA FROM 1 TAPE(DISK) AT A TIME                            FILEO    12
C                                                                       FILEO    13
       DO TO 10, FOR LL=(1)(10)                                         FILEO    14
       CALL PUTFR(X,LL,1)                                               FILEO    15
C                                                                       FILEO    16
CC     CREATE THE SET FRS FOR THE TAPE                                  FILEO    17
C                                                                       FILEO    18
     1 CALL GETFR(WWW,LL,IK)                                            FILEO    19
       IF IK NE 0, GO TO 2                                              FILEO    20
       CREATE FR                                                        FILEO    21
       LET W1(FR) = WWW(1)                                              FILEO    22
       LET W2(FR) = WWW(2)                                              FILEO    23
       LET W3(FR) = WWW(3)                                              FILEO    24
```

```
      LET W4(FR) = WWW(4)                                                FILEO     25
      FILE FR IN FRS                                                     FILEO     26
      GO TO 1                                                            FILEO     27
    2 IF FRS IS EMPTY, GO TO 10                                          FILEO     28
C                                                                        FILEO     29
CC    PROCESS THE SET FRS TO PRINT ALL SATELLITES ON THE TAPE            FILEO     30
C                                                                        FILEO     31
      DO TO 5, FOR ALL FR IN FRS                                         FILEO     32
      LET TIME = TIMEF(FR)                                               FILEO     33
      LET IS = SATNO(FR)                                                 FILEO     34
      LET I = SATSY(FR)                                                  FILEO     35
      LET NPOS(IS) = NPS(FR)                                             FILEO     36
      IF I EQ 1, LET K = UP                                              FILEO     37
      IF I EQ 2, LET K = DOWN                                            FILEO     38
      IF I EQ 3, LET K = OUT                                             FILEO     39
      LET STAT(ITSYS(IS)) = K                                            FILEO     40
      LET I = ST(FR)                                                     FILEO     41
      IF I EQ 1, LET K = UP                                              FILEO     42
      IF I EQ 2, LET K = DOWN                                            FILEO     43
      IF I EQ 3, LET K = OUT                                             FILEO     44
      LET SSTAT(IS) = K                                                  FILEO     45
      LET FREE = NDEL(FR)                                                FILEO     46
      IF INOW NE IS, WRITE ON 6                                          FILEO     47
      FORMAT(*0          CHRONOLOGICAL TIME HISTORY OF SATELLITE POSITION I FILEO     48
     *N ORBIT*/S5,*TIME              SYSTEM    STATUS    SATELLITE STATUS  FILEO     49
     * MODULE    STATUS*)                                                 FILEO     50
      LET INOW = IS                                                      FILEO     51
      CALL STATUS(IS,MODNO(FR),NOSTA(FR))                                FILEO     52
C                                                                        FILEO     53
CCC   RELEASE MEMORY                                                     FILEO     54
C                                                                        FILEO     55
      REMOVE FR FROM FRS                                                 FILEO     56
      DESTROY FR                                                         FILEO     57
    5 LOOP                                                               FILEO     58
   10 LOOP                                                               FILEO     59
      LET TRIG2 = 2                                                      FILEO     60
      RETURN                                                             FILEO     61
VINOW 0                                                                  FILEO     62
      END                                                                FILEO     63
      SUBROUTINE FILES(IS,IM,IST)                                        FILES      2
C                                                                        FILES      3
CCC   STORE SATELLITE DATA FOR THE SET FRS ON 10 TAPES ON DISK           FILES      4
CC    USE FR TEMPORARILY                                                 FILES      5
C                                                                        FILES      6
      DIMENSION WWW(4)                                                   FILES      7
      CREATE FR                                                          FILES      8
      LET TIMEF(FR) = TIME                                               FILES      9
      LET SATNO(FR) = IS                                                 FILES     10
      LET I = STAT(ITSYS(IS))                                            FILES     11
      IF I EQ UP, LET K = 1                                              FILES     12
      IF I EQ DOWN, LET K = 2                                            FILES     13
      IF I EQ OUT, LET K = 3                                             FILES     14
      LET SATSY(FR) = K                                                  FILES     15
      LET I = SSTAT(IS)                                                  FILES     16
      IF I EQ UP, LET K = 1                                              FILES     17
      IF I EQ DOWN, LET K = 2                                            FILES     18
      IF I EQ OUT, LET K = 3                                             FILES     19
```

OK

```
      LET ST(FR) = K                                                    FILES    20
      LET MODNO(FR) = IM                                                FILES    21
      LET NOSTA(FR) = IST                                               FILES    22
      LET NDEL(FR) = FREE                                               FILES    23
      LET NPS(FR) = NPOS(IS)                                            FILES    24
      LET LL = (10*IS+SYORB-1)/SYORB                                    FILES    25
      LET WWW(1) = W1(FR)                                               FILES    26
      LET WWW(2) = W2(FR)                                               FILES    27
      LET WWW(3) = W3(FR)                                               FILES    28
      LET WWW(4) = W4(FR)                                               FILES    29
      CALL PUTFR(WWW,LL,0)                                              FILES    30
      DESTROY FR                                                        FILES    31
      RETURN                                                            FILES    32
      END                                                               FILES    33
      SUBROUTINE GETV(IGO)                                              GETV     2
C                                                                       GETV     3
CC  FIND NECESSARY VEHICLES                                             GETV     4
C                                                                       GETV     5
      LET IPAD = 0                                                      GETV     6
      LET ITUG = 0                                                      GETV     7
      LET ISEPS = 0                                                     GETV     8
      LET IGO = 0                                                       GETV     9
      LET ISHUT = 0                                                     GETV     10
C                                                                       GETV     11
CCC   LOCATE NEXT AVAILABLE LAUNCH PAD                                  GETV     12
C                                                                       GETV     13
      DO TO 25, FOR I=(NPAD1(IORB))(NPAD2(IORB))                        GETV     14
      IF VPAD(I) LE 0, GO TO 25                                         GETV     15
      LET IPAD = I                                                      GETV     16
      GO TO 1                                                           GETV     17
   25 LOOP                                                              GETV     18
      LET IGO = 4                                                       GETV     19
      RETURN                                                            GETV     20
C                                                                       GETV     21
CCC   LOCATE NEXT AVAILABLE SHUTTLE IN FLEET                            GETV     22
C                                                                       GETV     23
    1 DO TO 5, FOR I=(1)(NSHUT)                                         GETV     24
      IF NQ LT 0, LET NLEG = 2                                          GETV     25
      IF VSHUT(I) LE 0, GO TO 5                                         GETV     26
      LET ISHUT = I                                                     GETV     27
      GO TO 6                                                           GETV     28
    5 LOOP                                                              GETV     29
      LET IGO = 1                                                       GETV     30
      RETURN                                                            GETV     31
C                                                                       GETV     32
CCC   LOCATE NEXT AVAILABLE UPPER STAGE IN FLEET                        GETV     33
C                                                                       GETV     34
    6 IF RQUP(IORB) EQ 0, GO TO 20                                      GETV     35
      DO TO 10, FOR I=(1)(NTUG)                                         GETV     36
      IF VTUG(I) LE 0, GO TO 10                                         GETV     37
      LET ITUG = I                                                      GETV     38
      GO TO 7                                                           GETV     39
   10 LOOP                                                              GETV     40
      LET IGO = 2                                                       GETV     41
      RETURN                                                            GETV     42
C                                                                       GETV     43
CC    LOCATE NEXT AVAILABLE SEPS IN FLEET                               GETV     44
```

```
C
      7 IF RQSEP(IORB) EQ 0, GO TO 20                                    GETV    45
        DO TO 15, FOR I=(1)(NSEPS)                                       GETV    46
        IF VSEPS(I) LE 0, GO TO 15                                       GETV    47
        LET ISEPS = I                                                    GETV    48
        GO TO 20                                                         GETV    49
     15 LOOP                                                             GETV    50
        LET IGO = 3                                                      GETV    51
     20 RETURN                                                           GETV    52
        END                                                              GETV    53
        SUBROUTINE ISPAY(WGH,WGHDN)                                      GETV    54
C                                                                        ISPAY    2
C       SET UP PAYLOAD ARRIVAL AND REMOVAL FROM ORBIT EVENT SEQUENCE     ISPAY    3
C                                                                        ISPAY    4
C       REIRIEVE LAUNCH DATA FROM LOADING QUEUE - PQUE AND CITEM         ISPAY    5
C                                                                        ISPAY    6
        IF NQ GT 0, GO TO 7                                              ISPAY    7
        IF ISEPS EQ 0, GO TO 7                                           ISPAY    8
        IF EXPV(RQSEP(IORB)) NE 0., GO TO 20                             ISPAY    9
      7 LET DUMMY = 0                                                    ISPAY   10
        LET FLYT = ORBTM(IORB)                                           ISPAY   11
        LET ILOAD(1) = PQUE(IORB)                                        ISPAY   12
        LET NQ = NL(IORB)                                                ISPAY   13
        LET ILOAD(J+1) = CITEM(ILOAD(J)), FOR J=(1)(NQ-1)                ISPAY   14
        LET NMD = ANMD(IORB)                                             ISPAY   15
        LET SU = (NMD+NINSU-1)/NINSU                                     ISPAY   16
        LET WGH = SU*WTSU                                                ISPAY   17
        LET WLEN = SU*LENSU                                              ISPAY   18
        LET WGHDN = 0.                                                   ISPAY   19
        LET WLEND = 0.                                                   ISPAY   20
        IF EXVEH EQ 0, LET WGHDN = WGH                                   ISPAY   21
        IF EXVEH EQ 0, LET WLEND = WLEN                                  ISPAY   22
        IF PSERV EQ 1, LET WGHDN = 0.                                    ISPAY   23
        IF PSERV EQ 1, LET WLEND = 0.                                    ISPAY   24
        DO TO 10, FOR I=(1)(NQ)                                          ISPAY   25
        LET NX = ILOAD(I)                                                ISPAY   26
        IF IRT(NX) NE 0, GO TO 12                                        ISPAY   27
        LET WGH = WGH + PAYWT(NX)                                        ISPAY   28
        LET WLEN = WLEN + PAYLN(NX)                                      ISPAY   29
        IF IMOD(NX) EQ 0, GO TO 10                                       ISPAY   30
     11 IF EXVEH NE 0, GO TO 10                                          ISPAY   31
        IF IMOD(NX) EQ 0, GO TO 12                                       ISPAY   32
        IF PSERV NE 0, GO TO 10                                          ISPAY   33
     12 LET WGHDN = WGHDN + PAYWT(NX)                                    ISPAY   34
        LET WLEND = WLEND + PAYLN(NX)                                    ISPAY   35
     10 LOOP                                                             ISPAY   36
        IF ISEPS EQ 0, GO TO 14                                          ISPAY   37
        IF NQ EQ -2, GO TO 150                                           ISPAY   38
        LET WGH = WGH + WUSEP                                            ISPAY   39
        IF WUSEP NE 0., LET WLEN = WLEN + LSEP                           ISPAY   40
        LET D = WGHDN                                                    ISPAY   41
        LET WGHDN = SWDN(ISEPS)                                          ISPAY   42
        LET SWDN(ISEPS) = D                                              ISPAY   43
        LET D = WLEND                                                    ISPAY   44
        LET WLEND = SLDN(ISEPS)                                          ISPAY   45
        LET SLDN(ISEPS) = D                                              ISPAY   46
        LET WGHDN = WGHDN + WDNSP                                        ISPAY   47
                                                                         ISPAY   48
```

```
      IF WDNSP NE 0., LET WLEND = WLEND + LSEP                          ISPAY     49
      GO TO 14                                                          ISPAY     50
150   LET WGHDN = WDNSP                                                 ISPAY     51
      LET WLEND = LSEP                                                  ISPAY     52
14    LET DUMMY = 0                                                     ISPAY     53
C                                                                       ISPAY     54
CC    DEFINE PAYLOADS IN LAUNCH                                         ISPAY     55
C                                                                       ISPAY     56
      IF TRIG EQ 0, WRITE ON 6,IPAD,ISHUT,ITUG,ISEPS,WGH,WGHDN,WLEN,    ISPAY     57
     * WLEND                                                            ISPAY     58
      FORMAT(S5,*--LAUNCH NOW-- PAD*,I2,* -- SHUTTLE*,I3,* -- TUG*,I3,*  ISPAY     59
     *-- SEPS*,I2,* -- WEIGHT =*,D6,*/*,D5,* -- LENGTH =*,D3.1,*/*,D2.2, ISPAY     60
     **--*)                                                             ISPAY     61
      IF TIME GT TIMEB, CALL CSPAY                                      ISPAY     62
      LET TP = PADT                                                     ISPAY     63
      LET T = 0.                                                        ISPAY     64
      IF ISEPS NE 0, LET T = TDOWN                                      ISPAY     65
      IF TP LT T, LET TP = T                                            ISPAY     66
      IF ISEPS EQ 0, GO TO 5                                            ISPAY     67
      IF TRIG NE 0, GO TO 5                                             ISPAY     68
      LET TE = TIME                                                     ISPAY     69
      LET I = DPART (TE)                                                ISPAY     70
      LET J = HPART (TE) + 1                                            ISPAY     71
      LET K = MPART (TE) + 1                                            ISPAY     72
      IF WUSEP NE 0., WRITE ON 6,I,J,K,ISEPS                            ISPAY     73
      FORMAT(*0     *,I5,*.*,I2,*.*,I2,S63,*SEPS  *,I3,* LAUNCHED*)     ISPAY     74
      IF WDNSP NE 0., WRITE ON 6,I,J,K,ISEPS                            ISPAY     75
      FORMAT(*0     *,I5,*.*,I2,*.*,I2,S63,*SEPS  *,I3,* RETRIEVED*)    ISPAY     76
5     LET DUMMY = 0                                                     ISPAY     77
      DO TO 17, FOR J=(1)(NL(IOR3))                                     ISPAY     78
      LET IK = ILOAD(J)                                                 ISPAY     79
      LET NX = ISAT(IK)                                                 ISPAY     80
      LET NY = IMOD(IK)                                                 ISPAY     81
      LET AST = SORTE(ITSAT(NX))                                        ISPAY     82
      IF AST NE 0, LET FLYT = AST                                       ISPAY     83
      IF IRT(IK) NE 0, GO TO 16                                         ISPAY     84
C                                                                       ISPAY     85
CC    DEPLOYMENT PAYLOADS                                               ISPAY     86
C                                                                       ISPAY     87
      LET FREE = LQTIM(IK)/3000.                                        ISPAY     88
      CALL STATUS(NX,NY,4)                                              ISPAY     89
      CREATE ARRIV                                                      ISPAY     90
      LET PSAT(ARRIV) = NX                                              ISPAY     91
      LET PMOD(ARRIV) = NY                                              ISPAY     92
      CAUSE ARRIV AT TIME + TP + GOTIM(IK)                              ISPAY     93
      IF AST EQ 0., GO TO 15                                            ISPAY     94
      LET GOTIM(IK) = AST                                               ISPAY     95
16    IF AST NE 0., GO TO 160                                           ISPAY     96
      CREATE BACK                                                       ISPAY     97
C                                                                       ISPAY     98
CC    SCHEDULE RETRIEVALS                                               ISPAY     99
C                                                                       ISPAY    100
      LET PSAT(BACK) = NX                                               ISPAY    101
      CAUSE BACK AT TIME + TP + FLYT                                    ISPAY    102
160   CREATE REMOV                                                      ISPAY    103
      LET PSAT(REMOV) = NX                                              ISPAY    104
      CAUSE REMOV AT TIME + TP + GOTIM(IK)                              ISPAY    105
```

```
      CREATE SATDN                                                      ISPAY    106
      LET PSAT(SATDN)=NX                                                ISPAY    107
      CAUSE SATDN AT TIME + IP + GOTIM(IK) - .01/8640.                  ISPAY    108
C                                                                       ISPAY    109
CC    REMOVE PAYLOAD FROM LOADING QUEUE                                 ISPAY    110
C                                                                       ISPAY    111
   15 CALL DROPQ(IK,IORB)                                               ISPAY    112
   17 LOOP                                                              ISPAY    113
   19 IF TRIG EQ 0, WRITE ON 6                                          ISPAY    114
      FORMAT(S5,*-------------------------------------------------      ISPAY    115
     *-------------------------------------------------*)               ISPAY    116
      LET NL(IORB) = 0                                                  ISPAY    117
      RETURN                                                            ISPAY    118
20    LET DUMMY = 0                                                     ISPAY    119
      IF TRIG NE 0, RETURN                                              ISPAY    120
      LET TE = TIME                                                     ISPAY    121
      LET I = DPART (TE)                                                ISPAY    122
      LET J = HPART (TE) + 1                                            ISPAY    123
      LET K = MPART (TE) + 1                                            ISPAY    124
      WRITE ON 6,I,J,K,ISEPS                                            ISPAY    125
      FORMAT(*0     *,I5,*.*,I2*.*,I2,S63,*SEPS  *I3,* EXPENDED*)       ISPAY    126
      RETURN                                                            ISPAY    127
      END                                                               ISPAY    128
      SUBROUTINE ISVEH(WGH,WGHDN)                                       ISVEH      2
C                                                                       ISVEH      3
CC    COLLECT STATISTICS ON VEHICLE UNAVAILABILITY                      ISVEH      4
C                                                                       ISVEH      5
      LET SEPEX = 0                                                     ISVEH      6
      IF NQ GT 0, GO TO 50                                              ISVEH      7
      IF ISEPS EQ 0, GO TO 50                                           ISVEH      8
      IF EXPV(RQSEP(IORB)) EQ 0., GO TO 50                              ISVEH      9
      LET SEPEX = 1                                                     ISVEH     10
      GO TO 180                                                         ISVEH     11
50    LET DUMMY = 0                                                     ISVEH     12
      DO TO 5, FOR I=(1)(4)                                             ISVEH     13
         GO TO (1,2,3,4),I                                              ISVEH     14
    1 IF ISHUT NE 0, GO TO 6                                            ISVEH     15
      GO TO 5                                                           ISVEH     16
    2 IF ITUG NE 0, GO TO 6                                             ISVEH     17
      GO TO 5                                                           ISVEH     18
    3 IF ISEPS NE 0, GO TO 6                                            ISVEH     19
      GO TO 5                                                           ISVEH     20
    4 IF IPAD EQ 0, GO TO 5                                             ISVEH     21
    6 IF VDATE(I) EQ 0., GO TO 5                                        ISVEH     22
      LET VDATE(I) = VDATE(I) + TIME                                    ISVEH     23
      IF VDATE(I) LT 0., GO TO 5                                        ISVEH     24
      LET VTD(I) = VTD(I) + VDATE(I)                                    ISVEH     25
      IF VDATE(I) GT XTD(I), LET XTD(I) = VDATE(I)                      ISVEH     26
      IF VDATE(I) LT MTD(I), LET MTD(I) = VDATE(I)                      ISVEH     27
      LET VDATE(I) = 0.                                                 ISVEH     28
    5 LOOP                                                              ISVEH     29
C                                                                       ISVEH     30
CC    SET UP EVENT SEQUENCE FOR VEHICLES                                ISVEH     31
CC    TO BECOME AVAILABLE AT A LATER TIME                               ISVEH     32
C                                                                       ISVEH     33
CC    SHUTTLE                                                           ISVEH     34
C                                                                       ISVEH     35
```

```
      LET TP = PADT                                                    ISVEH   36
      LET T = 0.                                                       ISVEH   37
      IF ISEPS NE 0, LET T = TDOWN                                     ISVEH   38
      IF TP LT T, LET TP = T                                           ISVEH   39
      LET TF = FLYT                                                    ISVEH   40
      IF ISEPS NE 0, LET TF = 12./8640.                                ISVEH   41
      CREATE REFVE                                                     ISVEH   42
      LET VNAME(REFVE) = SHUT                                          ISVEH   43
      LET PMOD(REFVE) = ISHUT                                          ISVEH   44
      CAUSE REFVE AT TIME + TP + SREFT + TF                            ISVEH   45
      LET VSHUT(ISHUT) = 0                                             ISVEH   46
      LET I = TIME - TIMEB + 1.                                        ISVEH   47
      IF I LE 0, GO TO 20                                              ISVEH   48
      LET SUTFY(I) = SUTFY(I) + 1                                      ISVEH   49
      IF ITUG NE 0, GO TO 20                                           ISVEH   50
      LET CSHUT(IORB) = CSHUT(IORB) + 1.                               ISVEH   51
      LET WSHUT(IORB) = WSHUT(IORB) + WGH                              ISVEH   52
      LET CDSUT(IORB) = CDSUT(IORB) + 1.                               ISVEH   53
      LET WDSUT(IORB) = WDSUT(IORB) + WGHDN                            ISVEH   54
C                                                                      ISVEH   55
CC    TUG                                                              ISVEH   56
C                                                                      ISVEH   57
   20 IF ITUG EQ 0, GO TO 18                                           ISVEH   58
      IF EXORB(IORB) NE 0, GO TO 22                                    ISVEH   59
      CREATE REFVE                                                     ISVEH   60
      LET VNAME(REFVE) = TUG                                           ISVEH   61
      LET PMOD(REFVE) = ITUG                                           ISVEH   62
      CAUSE REFVE AT TIME + TP + TREFT + TF                            ISVEH   63
   22 LET VTUG(ITUG) = 0                                               ISVEH   64
        IF I LE 0, GO TO 18                                            ISVEH   65
      LET TUGFY(I) = TUGFY(I) + 1                                      ISVEH   66
      IF ISEPS NE 0, GO TO 18                                          ISVEH   67
      LET EXVEH = EXORB(IORB)                                          ISVEH   68
      LET EXORB(IORB) = 0                                              ISVEH   69
      IF EXVEH EQ 0, LET EXVEH = EXPV(ROUP(IORB))                      ISVEH   70
      IF EXVEH NE 0, LET EXTUG = EXTUG + 1.                            ISVEH   71
      IF EXORB(IORB) NE 0, LET VTUG(ITUG) = -1                         ISVEH   72
      LET WTUG(IORB) = WTUG(IORB) + WGH                                ISVEH   73
      LET CTUG(IORB) = CTUG(IORB) + 1.                                 ISVEH   74
      IF EXORB(IORB) NE 0, GO TO 18                                    ISVEH   75
      LET CDTUG(IORB) = CDTUG(IORB) + 1.                               ISVEH   76
      LET WDTUG(IORB) = WDTUG(IORB) + WGHDN                            ISVEH   77
C                                                                      ISVEH   78
CC    SEPS/SCOOTER                                                     ISVEH   79
C                                                                      ISVEH   80
   18 IF ISEPS EQ 0, GO TO 19                                          ISVEH   81
      IF SEPEX NE 0, GO TO 180                                         ISVEH   82
      LET TS =0                                                        ISVEH   83
      IF NQ LT 0, LET TS = SEPFT                                       ISVEH   84
      CREATE REFVE                                                     ISVEH   85
      LET VNAME(REFVE) = SEPS                                          ISVEH   86
      LET PMOD(REFVE) = ISEPS                                          ISVEH   87
      CAUSE REFVE AT TIME + TP + FLYT                                  ISVEH   88
     * + TS                                                           ISVEH   89
  180 LET DUMMY = 0                                                    ISVEH   90
      LET VSEPS(ISEPS) = 0                                             ISVEH   91
      IF SEPEX NE 0, LET VSEPS(ISEPS) = -1                             ISVEH   92
```

```
      LET MSEP(ISEPS) = 0                                                ISVEH    93
      IF NQ LT 0, LET NEXIT(ISEPS) = 0                                   ISVEH    94
      IF NQ LT 0, LET LEXIT(ISEPS) = 0                                   ISVEH    95
      IF I LE 0, GO TO 19                                                ISVEH    96
      LET SEPFY(I) = SEPFY(I) +1                                         ISVEH    97
      LET CSEPS(IORB) = CSEPS(IORB) + 1.                                 ISVEH    98
      LET WSEPS(IORB) = WSEPS(IORB) + WGH                                ISVEH    99
      LET CDSEP(IORB) = CDSEP(IORB) + 1.                                 ISVEH   100
      LET WDSEP(IORB) = WDSEP(IORB) + WGHDN                              ISVEH   101
C                                                                        ISVEH   102
C     LAUNCH PAD                                                         ISVEH   103
C                                                                        ISVEH   104
   19 IF IPAD EQ 0, GO TO 21                                             ISVEH   105
      CREATE REFVE                                                       ISVEH   106
      LET VNAME(REFVE) =       KPAD                                      ISVEH   107
      LET PMOD (REFVE) = IPAD                                            ISVEH   108
      LET PSAT(REFVE) = IORB                                             ISVEH   109
      CAUSE REFVE AT TIME + TP + PREFT                                   ISVEH   110
      LET VPAD(IPAD) = 0                                                 ISVEH   111
   21 RETURN                                                             ISVEH   112
UKPAD PAD                                                                ISVEH   113
      END                                                                ISVEH   114
      ENDOGENOUS EVENT LAUNC                                             LAUNC     2
C                                                                        LAUNC     3
C     MANDATORY LAUNCH EVENT                                             LAUNC     4
C                                                                        LAUNC     5
C     THIS EVENT OCCURS WITH AN ACTUAL LAUNCH SCHEDULED WITH DELAYS.     LAUNC     6
C                                                                        LAUNC     7
C     IT SCHEDULES ARRIVAL IN ORBIT, VEHICLE REFURB CYCLE, MODULE AND    LAUNC     8
C                                                                        LAUNC     9
C     SAIELLITE RETRIEVAL WITH REFURB CYCLE                              LAUNC    10
C                                                                        LAUNC    11
C     PREDICT ABORTED LAUNCHES AND LOST PAYLOADS                         LAUNC    12
C                                                                        LAUNC    13
      LET IEVLA = IEVLA + 1                                              LAUNC    14
      LET IQ = LQEV(LAUNC)                                               LAUNC    15
      LET MLEV(IQ) = 0                                                   LAUNC    16
      DESTROY LAUNC                                                      LAUNC    17
      IF ISAT(IQ) EQ 0, RETURN                                           LAUNC    18
      LET IORB = ORBIT(ITSAT(ISAT(IQ)))                                  LAUNC    19
      IF ORBQ(IORB) IS EMPTY, RETURN                                     LAUNC    20
      REMOVE IQ FROM ORBQ(IORB)                                          LAUNC    21
      LET LQTIM(IQ) = PRIOR(ITSAT(ISAT(IQ)))                             LAUNC    22
      FILE IQ IN ORBQ(IORB)                                              LAUNC    23
      LET NL(IORB) = 0                                                   LAUNC    24
      CALL GETV(IGO)                                                     LAUNC    25
      IF W(IORB) GT 0., LET W(IORB) = -W(IORB)                           LAUNC    26
      IF IGO EQ 3, GO TO 5                                               LAUNC    27
      IF IGO NE 0, GO TO 10                                              LAUNC    28
    5 CALL SHIP(0,0)                                                     LAUNC    29
      RETURN                                                             LAUNC    30
   10 IF TRIG NE 0, GO TO 12                                             LAUNC    31
      LET TE = TIME                                                      LAUNC    32
      LET I = DPART (TE)                                                 LAUNC    33
      LET J = HPART (TE) + 1                                             LAUNC    34
      LET K = MPART (TE) + 1                                             LAUNC    35
      WRITE ON 6,I,J,K                                                   LAUNC    36
```

```
      FORMAT(S5,I5,*.*,I2,*.*,I2,S60,*PAYLOAD DUE TO GO - NO VEHICLE OR PAD*)   LAUNC    37
   12 LET CVA(IGO) = CVA(IGO) + 1.                                              LAUNC    38
      LET VDATE(IGO) = VDATE(IGO) - TIME                                        LAUNC    39
      RETURN                                                                    LAUNC    40
      END                                                                       LAUNC    41
      SUBROUTINE LDAT                                                           LAUNC    42
C                                                                               LDAT     2
CCC   LOAD DATA SUBROUTINE                                                      LDAT     3
C                                                                               LDAT     4
      WRITE ON 6                                                                LDAT     5
      FORMAT(*1      INPUT DATA*//)                                             LDAT     6
      LET IRFLG = 0                                                             LDAT     7
      CALL LDVEH(IRFLG)                                                         LDAT     8
      CALL LDORB(IRFLG)                                                         LDAT     9
      CALL LDMOD(IRFLG)                                                         LDAT    10
      CALL LDSAT(IRFLG)                                                         LDAT    11
      CALL LDSYS(IRFLG)                                                         LDAT    12
      CALL LDSCH(IRFLG)                                                         LDAT    13
      CALL LDME(IRFLG)                                                          LDAT    14
      CALL LDPUR                                                                LDAT    15
      IF IRFLG EQ 0, RETURN                                                     LDAT    16
      WRITE ON 6                                                                LDAT    17
      FORMAT(*0-------- RUN STOPPED DUE TO DATA ERROR --------*)                LDAT    18
      STOP                                                                      LDAT    19
      END                                                                       LDAT    20
      SUBROUTINE LDME(IRFLG)                                                    LDAT    21
C                                                                               LDME     2
CCC   MISSION EQUIPMENT UPGRADE INPUT ROUTINE                                   LDME     3
C                                                                               LDME     4
      DIMENSION IA(5),A(4)                                                      LDME     5
      WRITE ON 6                                                                LDME     6
      FORMAT(*  ME UPGRADE SCHEDULES INPUT *)                                   LDME     7
C                                                                               LDME     8
CCC   LOAD MISSION EQUIPMENT UPGRADE SCHEDULE                                   LDME     9
C                                                                               LDME    10
  100 READ FROM 5,IA(1),IA(2),IA(3),IA(4),B,IA(5)                               LDME    11
      FORMAT(A6,I4,A6,I4,M4.2.2,A6)                                             LDME    12
C                                                                               LDME    13
CCC   PRINT SCHEDULES                                                           LDME    14
C                                                                               LDME    15
      WRITE ON 6,IA(1),IA(2),IA(3),IA(4),B,IA(5)                                LDME    16
      FORMAT(S10,A6,I6,S3,A6,I6,S3,M4.2.2,S3,A6)                                LDME    17
            IF IA(1) EQ BLANK, GO TO 200                                        LDME    18
      LET MEOLD = 0                                                             LDME    19
      LET MENEW = 0                                                             LDME    20
      DO TO 110, FOR I=(1)(MITAB)                                               LDME    21
      IF IA(3) EQ MNAME(I), LET MEOLD = I                                       LDME    22
      IF IA(5) EQ MNAME(I), LET MENEW = I                                       LDME    23
  110 LOOP                                                                      LDME    24
      IF MEOLD + MENEW NE 0, GO TO 115                                          LDME    25
C                                                                               LDME    26
CCC   ERROR DETECTED                                                            LDME    27
C                                                                               LDME    28
  111 WRITE ON 6                                                                LDME    29
      FORMAT(* BAD ME DATA - ENTRY REJECTED *)                                  LDME    30
      LET RTFLG = 1                                                             LDME    31
                                                                                LDME    32
```

```
      GO TO 100                                                          LDME      33
  115 IF MCLAS(MEOLD) NE ME, GO TO 111                                   LDME      34
      IF MCLAS(MENEW) NE ME, GO TO 111                                   LDME      35
      DO TO 120, FOR I=(1)(STST8)                                        LDME      36
      IF IA(1) NE SYNAM(I), GO TO 120                                    LDME      37
      LET ISY = I                                                        LDME      38
      GO TO 125                                                          LDME      39
  120 LOOP                                                               LDME      40
      GO TO 111                                                          LDME      41
  125 IF FSAT(ISY) EQ 0, GO TO 111                                       LDME      42
      LET ISY = FSAT(ISY)+IA(2) - 1                                      LDME      43
      IF MOD(ISY) IS EMPTY, GO TO 111                                    LDME      44
      DO TO 130, FOR ALL MODSY IN MOD(ISY)                               LDME      45
      IF NOMOD(MODSY) EQ MEOLD, LET IA(4) = IA(4) - 1                    LDME      46
      IF IA(4) EQ 0, GO TO 135                                           LDME      47
  130 LOOP                                                               LDME      48
      GO TO 111                                                          LDME      49
C                                                                        LDME      50
CC    SAVE ME UPGRADE IN MENEW                                           LDME      51
C                                                                        LDME      52
  135 CREATE MESET                                                       LDME      53
      LET PSAT(MESET) = ISY                                              LDME      54
      LET PMOD(MESET) = NOMOD(MODSY)                                     LDME      55
      LET  MEDT(MESET)= 8                                                LDME      56
      LET NOMOD(MESET) = MENEW                                           LDME      57
      FILE MESET IN MES                                                  LDME      58
      GO TO 100                                                          LDME      59
  200 RETURN                                                             LDME      60
UME   ME                                                                 LDME      61
      END                                                                LDME      62
      SUBROUTINE LDMOD(IRFLG)                                            LDMOD      2
C                                                                        LDMOD      3
CC    MODULE INPUT ROUTINE                                               LDMOD      4
C                                                                        LDMOD      5
      READ FROM 5,NUMMOD,FACT                                            LDMOD      6
      FORMAT(I3,D1.3)                                                    LDMOD      7
      IF NUMMOD LE MITAB, GO TO 5                                        LDMOD      8
      WRITE ON 6,NUMMOD,MITAB                                            LDMOD      9
      FORMAT(*  ERROR - NUMBER OF MODULES INPUT(*,I6,*) EXCEEDS CAPACITY LDMOD     10
     *(*,I6,*)*)                                                         LDMOD     11
      LET IRFLG = 1                                                      LDMOD     12
    5 WRITE ON 6,NUMMOD                                                  LDMOD     13
      FORMAT(I11,* MODULES INPUT*/*     NAME          ALPHA F    BETA F  LDMOD     14
     *   T TIME    ALPHA W    BETA W    WEIGHT    VOLUME    CLASS*)      LDMOD     15
      DO TO 10, FOR I=(1)(NUMMOD)                                        LDMOD     16
C                                                                        LDMOD     17
CC    LOAD MODULE DATA                                                   LDMOD     18
C                                                                        LDMOD     19
      READ FROM 5, MNAME(I)                                              LDMOD     20
     * ,ALPF(I),BETAF(I),TTFMO(I),MODWT(I),MDVOL(I),                     LDMOD     21
     *  MCLAS(I)                                                         LDMOD     22
     *,ALPW(I),BETAW(I)                                                  LDMOD     23
     *,R,TAU                                                             LDMOD     24
      FORMAT(A6,D6.2,D2.2,D3,D5,D3.1,A6,D5.2,D2.2,D1,D2.2)               LDMOD     25
      IF ALPF(I) NE 0., GO TO 7                                          LDMOD     26
      IF R EQ 0., GO TO 7                                                LDMOD     27
      LET BETAF(I) = 1.                                                  LDMOD     28
```

```
      LET ALPF(I) = -TAU/ALOG(R)                                        LDMOD    29
    7 IF ALPW(I) EQ 0., LET ALPW(I) = FACT*ALPF(I)                      LDMOD    30
      IF BETAW(I) EQ 0., LET BETAW(I) = 1.                              LDMOD    31
      IF TTFMD(I) EQ 0., LET TTFMD(I)=.5*ALPF(I)                        LDMOD    32
C                                                                       LDMOD    33
CC    PRINT MODULE DATA                                                 LDMOD    34
C                                                                       LDMOD    35
      WRITE ON 6,RNAME(I),ALPF(I),BETAF(I),TTFMD(I),ALPW(I),BETAW(I),   LDMOD    36
     *  MODWT(I),MDVOL(I),MCLAS(I)                                      LDMOD    37
      FORMAT(S5,A6,S4,7D7.2,S4,A6)                                      LDMOD    38
   10 LOOP                                                              LDMOD    39
      RETURN                                                            LDMOD    40
      END                                                               LDMOD    41
      SUBROUTINE LDORB(IRFLG)                                           LDORB     2
C                                                                       LDORB     3
CC    LOAD ORBIT DATA                                                   LDORB     4
C                                                                       LDORB     5
      READ FROM 5,NORB                                                  LDORB     6
      FORMAT(I3)                                                        LDORB     7
      IF NORB LE NORBS, GO TO 1                                         LDORB     8
      WRITE ON 6,NORB,NORBS                                             LDORB     9
      FORMAT(*  ERROR - NUMBER OF ORBITS INPUT(*,I6,*) EXCEEDS CAPACITY( LDORB    10
     **,I6,*)*)                                                         LDORB    11
      LET IRFLG = 1                                                     LDORB    12
    1 WRITE ON 6,NORB                                                   LDORB    13
      FORMAT(I8,* ORBITS INPUT*)                                        LDORB    14
      WRITE ON 6                                                        LDORB    15
      FORMAT(*   NAME        DV     PERIOD     RA      VC  UPPER  SEPS  LDORB    16
     *  SHUTTLE      DV1       PADS*)                                   LDORB    17
      DO TO 10, FOR I=(1)(NORB)                                         LDORB    18
      READ FROM 5,ORBID(I),ORBDV(I),ORBPD(I),ORBRA(I),ORBVC(I),RQUP(I), LDORB    19
     * RQSEP(I),RQSUT(I),DV1(I)                                        LDORB    20
     *,NPAD1(I),NPAD2(I)                                               LDORB    21
      FORMAT(A6,4D5.1,3A6,D5.1,2I3)                                     LDORB    22
      IF NPAD1(I) EQ 0, LET NPAD1(I) = 1                                LDORB    23
      IF NPAD1(I) GT NPAD, LET NPAD1(I) = NPAD                          LDORB    24
      IF NPAD2(I) EQ 0, LET NPAD2(I) = NPAD                             LDORB    25
      IF NPAD2(I) GT NPAD, LET NPAD2(I) = NPAD                          LDORB    26
      WRITE ON 6 ,ORBID(I),ORBDV(I),ORBPD(I),ORBRA(I),ORBVC(I),RQUP(I), LDORB    27
     * RQSEP(I),RQSUT(I),DV1(I)                                        LDORB    28
     *,NPAD1(I),NPAD2(I)                                               LDORB    29
      FORMAT(S3,A6,4D7.1,S1,A6,S1,A6,S1,A6,D7.1,S4,2I3)                 LDORB    30
C                                                                       LDORB    31
CC    CHECK ON UPPER STAGE                                              LDORB    32
C                                                                       LDORB    33
      LET J = 0                                                         LDORB    34
      IF RQUP(I) EQ BLANK, GO TO 9                                      LDORB    35
      DO TO 5, FOR J=(1)(NVEH)                                          LDORB    36
      IF RQUP(I) EQ NAMEV(J), GO TO 9                                   LDORB    37
    5 LOOP                                                              LDORB    38
      LET IRFLG = 1                                                     LDORB    39
      WRITE ON 6                                                        LDORB    40
      FORMAT(*  NO SUCH UPPER STAGE*)                                   LDORB    41
    9 LET RQUP(I) = J                                                   LDORB    42
C                                                                       LDORB    43
CC    CHECK ON SEPS VEHICLE                                             LDORB    44
C                                                                       LDORB    45
```

```
      LET J= 0                                                          LDORB   46
      IF RQSEP(I) EQ BLANK, GO TO 4                                     LDORB   47
    2 DO TO 3, FOR J=(1)(NVEH)                                          LDORB   48
      IF RQSEP(I) EQ NAMEV(J), GO TO 4                                  LDORB   49
    3 LOOP                                                              LDORB   50
      LET IRFLG = 1                                                     LDORB   51
      WRITE ON 6                                                        LDORB   52
      FORMAT(*  NO SUCH SEPS VEHICLE FOUND *)                           LDORB   53
    4 LET RQSEP(I) = J                                                  LDORB   54
      IF NAMEV(J) NE SEPS, LET CHEM = 1                                 LDORB   55
      IF CHEM NE 0, CALL LDSEP(WOV(J),PAYLV(J),WCONV(J),ISPV(J),        LDORB   56
     * WPNOV(J),EXPV(J),DAYSV(J),REFTV(J))                             LDORB   57
      LET SEPEX = EXPV(J)                                               LDORB   58
C                                                                       LDORB   59
CC    CHECK ON SHUTTLE VEHICLE                                          LDORB   60
C                                                                       LDORB   61
      LET J = 0                                                         LDORB   62
    6 DO TO 7, FOR J=(1)(NVEH)                                          LDORB   63
      IF RQSUT(I) EQ NAMEV(J), GO TO 8                                  LDORB   64
    7 LOOP                                                              LDORB   65
      LET IRFLG = 1                                                     LDORB   66
      WRITE ON 6                                                        LDORB   67
      FORMAT(*  NO SUCH SHUTTLE FOUND *)                                LDORB   68
    8 LET RQSUT(I) = J                                                  LDORB   69
   10 LOOP                                                              LDORB   70
      RETURN                                                            LDORB   71
      END                                                               LDORB   72
      SUBROUTINE LDPUR                                                  LDPUR   2
C                                                                       LDPUR   3
CC  PURGE MEMORY OF UNUSED MODULES                                      LDPUR   4
C                                                                       LDPUR   5
      WRITE ON 6                                                        LDPUR   6
      FORMAT(*1      SYNOPSIS OF INPUT*)                                LDPUR   7
      LET K = 0                                                         LDPUR   8
      LET M = 0                                                         LDPUR   9
      DO TO 80, FOR I=(1)(STSTB)                                        LDPUR   10
      LET NSYLF(I) = 1000.                                              LDPUR   11
      LET J = 0                                                         LDPUR   12
      IF FSAT(I) EQ 0, GO TO 80                                         LDPUR   13
      DO TO 79, FOR L=(FSAT(I))(LSAT(I))                                LDPUR   14
      IF MARKS(L) EQ 0, GO TO 79                                        LDPUR   15
      LET MARKS(L) = 0                                                  LDPUR   16
      LET J = 1                                                         LDPUR   17
      LET NMODS(ITSAT(L)) = 1                                           LDPUR   18
      LET MDCNT(NOMOD(MDSAT)) = 1, FOR ALL MDSAT IN MDS(ITSAT(L))       LDPUR   19
   79 LOOP                                                              LDPUR   20
      IF J NE 0, GO TO 78                                               LDPUR   21
      WRITE ON 6,SYNAM(I)                                               LDPUR   22
      FORMAT(* UNUSED SYSTEM - *,A6)                                    LDPUR   23
      LET SYNAM(I) = 0                                                  LDPUR   24
      GO TO 80                                                          LDPUR   25
   78 LET K = K + 1                                                     LDPUR   26
      LET M = M + LSAT(I) - FSAT(I) + 1                                 LDPUR   27
   80 LOOP                                                              LDPUR   28
      LET I = M/4                                                       LDPUR   29
      IF I*4 NE M, LET I = I+1                                          LDPUR   30
      LET M = I*4                                                       LDPUR   31
```

```
      WRITE ON 6,M,SYORB                                                LDPUR     32
      FORMAT(* PROBLEM USED *,I3,* SATELLITE/SYSTEM POSITIONS OUT OF AVA LDPUR     33
     *ILABLE *,I3)                                                      LDPUR     34
      WRITE ON 6,K,STSTB                                                LDPUR     35
      FORMAT(* PROBLEM USED *,I3,* SYSTEMS OUT OF AVAILABLE *,I3)       LDPUR     36
      LET K=0                                                           LDPUR     37
      DO TO 85, FOR I=(1)(SITAB)                                        LDPUR     38
      IF NMODS(I) NE 0, LET K = K+1                                     LDPUR     39
      IF NMODS(I) NE 0, GO TO 85                                        LDPUR     40
      IF MDS(I) IS EMPTY, GO TO 84                                      LDPUR     41
      DO TO 83, FOR ALL MDSAT IN MDS(I)                                 LDPUR     42
      REMOVE FIRST MDSAT FROM MDS(I)                                    LDPUR     43
      DESTROY MDSAT                                                     LDPUR     44
   83 LOOP                                                              LDPUR     45
   84 IF SNAME(I) EQ 0, GO TO 85                                        LDPUR     46
      WRITE ON 6,SNAME(I)                                               LDPUR     47
      FORMAT(*   UNUSED SATELLITE - *,A6)                              LDPUR     48
      LET SNAME(I) = 0                                                  LDPUR     49
   85 LOOP                                                              LDPUR     50
      WRITE ON 6,K,SITAB                                                LDPUR     51
      FORMAT(* PROBLEM USED *,I3,* SATELLITES OUT OF AVAILABLE *,I3)    LDPUR     52
      LET K = 0                                                         LDPUR     53
      DO TO 90, FOR I=(1)(MITAB)                                        LDPUR     54
      IF MNAME(I) EQ 0, GO TO 90                                        LDPUR     55
      IF MDCNT(I) NE 0, LET K = K + 1                                   LDPUR     56
      IF MDCNT(I) EQ 0, WRITE ON 6,MNAME(I)                             LDPUR     57
      FORMAT(*   UNUSED MODULE - *,A6)                                 LDPUR     58
      IF MDCNT(I) EQ 0, LET MNAME(I) = 0                                LDPUR     59
      LET MDCNT(I) = 0                                                  LDPUR     60
   90 LOOP                                                              LDPUR     61
      DO TO 6, FOR I=(1)(SYORB)                                         LDPUR     62
      IF ITSAT(I) EQ BLANK, GO TO 6                                     LDPUR     63
      IF ITSAT(I) EQ 0, GO TO 6                                         LDPUR     64
      IF MDS(ITSAT(I)) IS EMPTY, GO TO 6                                LDPUR     65
      LET J = 0                                                         LDPUR     66
      DO TO 4, FOR ALL MDSAT IN MDS(ITSAT(I))                           LDPUR     67
      CREATE MODSY                                                      LDPUR     68
      LET NOMOD(MODSY) = NOMOD(MDSAT)                                   LDPUR     69
      LET NUM  (MODSY) = 0                                              LDPUR     70
      LET SUMNU(MODSY) = 0                                              LDPUR     71
      LET MAXNU(MODSY) = 0                                              LDPUR     72
      LET MINNU(MODSY) = 500                                            LDPUR     73
      LET LOADF(MODSY) = 0                                              LDPUR     74
      LET SUMLF(MODSY) = 0                                              LDPUR     75
      LET MAXLF(MODSY) = 0                                              LDPUR     76
      LET MINLF(MODSY) = 1000                                           LDPUR     77
      LET MSTAT(MODSY) = 0                                              LDPUR     78
      LET NRU  (MODSY) = NRU(MDSAT)                                     LDPUR     79
      LET J = J + 1                                                     LDPUR     80
      LET MNO(MODSY) = J                                                LDPUR     81
      FILE MODSY IN MOD(I)                                              LDPUR     82
    4 LOOP                                                              LDPUR     83
    6 LOOP                                                              LDPUR     84
      WRITE ON 6,K,MITAB                                                LDPUR     85
      FORMAT(* PROBLEM USED *,I3,* MODULES OUT OF AVAILABLE *,I3)       LDPUR     86
      RETURN                                                            LDPUR     87
      END                                                               LDPUR     88
```

```
      SUBROUTINE LDSAT(IRFLG)                                                  LDSAT     2
C                                                                              LDSAT     3
CC    SATELLITE INPUT ROUTINE                                                  LDSAT     4
C                                                                              LDSAT     5
      DIMENSION IA(7),MODUL(7)                                                 LDSAT     6
      READ FROM 5,NUMSAT                                                       LDSAT     7
      FORMAT(I3)                                                               LDSAT     8
      IF NUMSAT LE SITA3, GO TO 6                                              LDSAT     9
      WRITE ON 6,NUMSAT,SITAB                                                  LDSAT    10
      FORMAT(*  ERROR - NUMBER OF SATELLITES INPUT(*,I6,*) EXCEEDS CAPAC       LDSAT    11
     *ITY(*,I6*) *)                                                            LDSAT    12
      LET IRFLG = 1                                                            LDSAT    13
    6 WRITE ON 6,NUMSAT                                                        LDSAT    14
      FORMAT(/S1,I10,*  SATELLITES INPUT*/*  NAME        WT      VOL  PRIO    LDSAT    15
     *R    INC        ORBIT   MOD        SAT TT   POLICY SORT   EXWT*)        LDSAT    16
      DO TO 25,  FOR I=(1)(NUMSAT)                                             LDSAT    17
         LET KL = 0                                                            LDSAT    18
C                                                                              LDSAT    19
CC    LOAD SATELLITE DATA                                                      LDSAT    20
C                                                                              LDSAT    21
      READ FROM 5,SNAME(I),SWT(I),SVOL(I),                                     LDSAT    22
     * PRIOR(I),INCL(I),ORBIT(I),NO                                            LDSAT    23
     *,TTSAT(I),POLDN(I)                                                       LDSAT    24
     *,SORTE(I),EXWT(I)                                                        LDSAT    25
      FORMAT(A6,S3,D5,D2.2,2D4,A6,S34,I5,D4,/I1,D4,D5)                         LDSAT    26
      IF EXWT(I) EQ 0., LET EXWT(I)=SWT(I)                                     LDSAT    27
      IF TTSAT(I) EQ 0., LET TTSAT(I)=10.                                      LDSAT    28
      IF PRIOR(I) EQ 0, LET PRIOR(I) = 6                                       LDSAT    29
C                                                                              LDSAT    30
CC    PRINT SATELLITE DATA                                                     LDSAT    31
C                                                                              LDSAT    32
      WRITE ON 6,SNAME(I),SWT(I),SVOL(I),PRIOR(I),INCL(I),ORBIT(I),NO          LDSAT    33
     *,TTSAT(I),POLDN(I)                                                       LDSAT    34
     *,SORTE(I),EXWT(I)                                                        LDSAT    35
      FORMAT (S2,A6,S1,4D6,S8,A6,I5,S7,D6,I8,D6,D6)                            LDSAT    36
      LET SORTE(I) = SORTE(I)/360.                                             LDSAT    37
      DO TO 1, FOR J=(1)(NORBS)                                                LDSAT    38
      IF ORBIT(I) NE ORBID(J), GO TO 1                                         LDSAT    39
      LET ORBIT(I) = J                                                         LDSAT    40
      GO TO 2                                                                  LDSAT    41
    1 LOOP                                                                     LDSAT    42
C                                                                              LDSAT    43
CC    ERROR DETECTED                                                           LDSAT    44
C                                                                              LDSAT    45
      LET IRFLG = 1                                                            LDSAT    46
      WRITE ON 6                                                               LDSAT    47
      FORMAT(*  ERROR - UNKNOWN ORBIT  *)                                      LDSAT    48
    2 LET J = 1                                                                LDSAT    49
C                                                                              LDSAT    50
CC    READ MODULE LIST FOR SATELLITE                                           LDSAT    51
C                                                                              LDSAT    52
      READ FROM 5,MODUL(J),IA(J),MODUL(J+1),IA(J+1),MODUL(J+2),IA(J+2),        LDSAT    53
     *        MODUL(J+3),IA(J+3),MODUL(J+4),IA(J+4),MODUL(J+5),IA(J+5),        LDSAT    54
     *        MODUL(J+6),IA(J+6)                                               LDSAT    55
      FORMAT(S10,A6,A4,A6,A4,A6,A4,A6,A4,A6,A4,A6,A4,A6,A4)                    LDSAT    56
C                                                                              LDSAT    57
CC    PRINT MODULE LIST                                                        LDSAT    58
```

```
C
      WRITE ON 6,MODUL(J),IA(J),MODUL(J+1),IA(J+1),MODUL(J+2),IA(J+2),
     *      MODUL(J+3),IA(J+3),MODUL(J+4),IA(J+4),MODUL(J+5),IA(J+5),
     *      MODUL(J+6),IA(J+6)
      FORMAT(S10,14A6)
      IF MODUL(1) NE LAST, GO TO 15
      LET NO = KL
      GO TO 25
15    LET DUMMY = 0
      DO TO 10, FOR J=(1)(7)
      IF MODUL(J) EQ BLANK, GO TO 10
      LET KL = KL + 1
      DO TO 20, FOR L=(1)(MITAB)
      IF MODUL(J) EQ MNAME(L), GO TO 5
   20 LOOP
C
CCC   ERROR DETECTED
C
      WRITE ON 6,MODUL(J)
      FORMAT(S3,* ERROR  MODULE - *,A6,* = NOT FOUND IN MODULE TABLE*)
      LET IRFLG = 1
      GO TO 10
C
CCC   PUT MODULE IN SET MDS BELONGING TO SATELLITE I
C
    5 CREATE MDSAT
      CALL CON(IA(J),K)
      LET NRU(MDSAT) = K
      LET NOMOD(MDSAT) = L
      FILE MDSAT IN MDS(I)
   10 LOOP
      IF NO EQ 0, GO TO 2
      IF KL LT NO, GO TO 2
      IF KL EQ NO, GO TO 25
      LET IRFLG = 1
      WRITE ON 6,NO,KL
      FORMAT(S3,*ERROR IN MODULE COUNT - EXPECTED *,I3,* FOUND *,I3)
   25 LOOP
      RETURN
VLAST LAST
      END
      SUBROUTINE LDSCH(IRFLG)
C
CCC   SATELLITE SCHEDULE INPUT ROUTINE
C
      DIMENSION IA(4),A(4),IB(4)
      WRITE ON 6
      FORMAT(*      SCHEDULES INPUT*)
C
CCC   LOAD SCHEDULES
C
   60 READ FROM 5,IA(1),IB(1),A(1),IA(2),IB(2),A(2),IA(3),IB(3),A(3),
     *IA(4),IB(4),A(4)
      FORMAT(I1,A6,S3,D4.5,I1,A6,S3,D4.5,I1,A6,S3,D4.5,I1,A6,S3,D4.5)
C
CCC   PRINT SCHEDULES
C
```

```
LDSAT   59
LDSAT   60
LDSAT   61
LDSAT   62
LDSAT   63
LDSAT   64
LDSAT   65
LDSAT   66
LDSAT   67
LDSAT   68
LDSAT   69
LDSAT   70
LDSAT   71
LDSAT   72
LDSAT   73
LDSAT   74
LDSAT   75
LDSAT   76
LDSAT   77
LDSAT   78
LDSAT   79
LDSAT   80
LDSAT   81
LDSAT   82
LDSAT   83
LDSAT   84
LDSAT   85
LDSAT   86
LDSAT   87
LDSAT   88
LDSAT   89
LDSAT   90
LDSAT   91
LDSAT   92
LDSAT   93
LDSAT   94
LDSAT   95
LDSAT   96
LDSAT   97
LDSAT   98
LDSAT   99
LDSCH    2
LDSCH    3
LDSCH    4
LDSCH    5
LDSCH    6
LDSCH    7
LDSCH    8
LDSCH    9
LDSCH   10
LDSCH   11
LDSCH   12
LDSCH   13
LDSCH   14
LDSCH   15
LDSCH   16
LDSCH   17
```

```
      WRITE ON 6,IA(1),IB(1),A(1),IA(2),IB(2),A(2),IA(3),IB(3),A(3),      LDSCH    18
     $IA(4),IB(4),A(4)                                                    LDSCH    19
      FORMAT(I6S2,,A6,S3,D4.5,I2,S2,A6,S3,D4.5,I2,S2,A6,S2,A6,S3,D4.5,I2,S2,A6 LDSCH  20
     *,S3,D4.5)                                                           LDSCH    21
      IF IA(1) EQ 0, GO TO 70                                             LDSCH    22
                                                                          LDSCH    23
C     FIND SYSTEM AND SAVE NEW SATELLITE LAUNCH IN NEWS                   LDSCH    24
C                                                                         LDSCH    25
      DO TO 65, FOR K = (1)(4)                                            LDSCH    26
      IF IA(K) EQ 0, GO TO 65                                             LDSCH    27
      IF A(K) GT TIMES, GO TO 65                                          LDSCH    28
      DO TO 66, FOR I=(1)(STST3)                                          LDSCH    29
      IF IB(K) NE SYNAM(I),GO TO 66                                       LDSCH    30
      LET J = LSAT(I) - FSAT(I) + 1                                       LDSCH    31
      IF IA(K) GT J, GO TO 64                                             LDSCH    32
                                                                          LDSCH    33
C     SCHEDULE INPUT DATA MATCHED WITH PREVIOUS DATA                      LDSCH    34
C                                                                         LDSCH    35
      LET MARKS(FSAT(I)-1+IA(K)) = 1                                      LDSCH    36
      CREATE NEW                                                          LDSCH    37
      LET SCHDT(NEW) = A(K)                                               LDSCH    38
      LET SCHSY(NEW) = FSAT(I)-1+IA(K)                                    LDSCH    39
      FILE NEW IN NEWS                                                    LDSCH    40
      GO TO 65                                                            LDSCH    41
                                                                          LDSCH    42
C     ERROR DETECTED                                                      LDSCH    43
C                                                                         LDSCH    44
   64 LET IRFLG = 1                                                       LDSCH    45
      WRITE ON 6,IA(K),IB(K)                                              LDSCH    46
      FORMAT(*   ERROR - MEMBER NO.*,I3,* IS NOT IN SYSTEM - *,A6)       LDSCH    47
      GO TO 65                                                            LDSCH    48
   66 LOOP                                                                LDSCH    49
                                                                          LDSCH    50
C     ERROR DETECTED                                                      LDSCH    51
C                                                                         LDSCH    52
      LET IRFLG = 1                                                       LDSCH    53
      WRITE ON 6,IB(K)                                                    LDSCH    54
      FORMAT(S3,* ERROR  SYSTEM NOT FOUND - *,A6)                         LDSCH    55
   65 LOOP                                                                LDSCH    56
      GO TO 60                                                            LDSCH    57
   70 RETURN                                                              LDSCH    58
      END                                                                 LDSCH    59
      SUBROUTINE LDSYS(IRFLG)                                             LDSYS     2
C                                                                         LDSYS     3
C     SYSTEMS INPUT ROUTINE                                               LDSYS     4
C                                                                         LDSYS     5
      READ FROM 5,NUMSYS                                                  LDSYS     6
      FORMAT(I3)                                                          LDSYS     7
      IF NUMSYS LE STST3, GO TO 1                                         LDSYS     8
      WRITE ON 6,NUMSYS,STST3                                             LDSYS     9
      FORMAT(*   ERROR - NUMBER OF SYSTEMS INPUT(*,I6*) EXCEEDS CAPACITY( LDSYS    10
     **,I6,*)*)                                                           LDSYS    11
      LET IRFLG = 1                                                       LDSYS    12
    1 WRITE ON 6,NUMSYS                                                   LDSYS    13
      FORMAT(/I11,*SYSTEMS INPUT*/*    NAME    NUP  NTOT   SYS TT    SAT LDSYS    14
     *     PHASE    SAT    PHASE    SAT    PHASE*)                       LDSYS    15
      LET J = 0                                                           LDSYS    16
```

```
      DO TO 60, FOR I=(1)(NUMSYS)                                        LDSYS     17
      LET FSAT(I) = J + 1                                                LDSYS     18
C                                                                        LDSYS     19
CCC   LOAD SATELLITE SYSTEMS DATA                                        LDSYS     20
C                                                                        LDSYS     21
      READ FROM 5,SYNAM(I),NFUP(I),NO,TTSYS(I),                          LDSYS     22
     * ITSAT(J+1),PHASE(J+1),ITSAT(J+2),                                 LDSYS     23
     * PHASE(J+2),ITSAT(J+3),PHASE(J+3)                                  LDSYS     24
      FORMAT(A6,2I5,D2.1,A6,S4,D4.5,A6,S4,D4.5,A6,S4,D4.5)               LDSYS     25
      IF TTSYS(I) EQ 0., LET TTSYS(I)=15.                                LDSYS     26
      IF NFUP(I) LE 0, LET NFUP(I) = 1                                   LDSYS     27
      IF NO LE 0, LET NO = 1                                             LDSYS     28
C                                                                        LDSYS     29
CCC   PRINT SATELLITE SYSTEMS DATA                                       LDSYS     30
C                                                                        LDSYS     31
      WRITE ON 6,SYNAM(I),NFUP(I),NO,TTSYS(I),                           LDSYS     32
     * ITSAT(J+1),PHASE(J+1)                                             LDSYS     33
     * ,ITSAT(J+2),PHASE(J+2),ITSAT(J+3),PHASE(J+3)                      LDSYS     34
      FORMAT(S2,A6,2I5,D6.2,S4,A6,D6.1,S4,A6,D6.1,S4,A6,D6.1)            LDSYS     35
      LET NSAT(I) = NO                                                   LDSYS     36
      DO TO 2, FOR J1=(1)(NO-3)(3)                                       LDSYS     37
      LET J2 = J1 + J - 1                                                LDSYS     38
      READ FROM 5,ITSAT(J2+4),PHASE(J2+4),ITSAT(J2+5),PHASE(J2+5)        LDSYS     39
     * ,ITSAT(J2+6),PHASE(J2+6)                                          LDSYS     40
      FORMAT(S20,A6,S4,D4.5,A6,S4,D4.5,A6,S4,D4.5)                       LDSYS     41
      WRITE ON 6,ITSAT(J2+4),PHASE(J2+4),ITSAT(J2+5),PHASE(J2+5)         LDSYS     42
     * ,ITSAT(J2+6),PHASE(J2+6)                                          LDSYS     43
      FORMAT(S31,A6,D6.1,S4,A6,D6.1,S4,A6,D6.1)                          LDSYS     44
    2 LOOP                                                               LDSYS     45
C                                                                        LDSYS     46
CCC   FIND SATELLITE                                                     LDSYS     47
C                                                                        LDSYS     48
    5 LET J = J + NO                                                     LDSYS     49
      LET LSAT(I) = J                                                    LDSYS     50
      DO TO 55, FOR L = (FSAT(I))(LSAT(I))                               LDSYS     51
      IF PHASE(L) LT 0., LET PHASE(L) = PHASE(L) + 360.                  LDSYS     52
      LET A = L                                                          LDSYS     53
      LET PHASE(L) = PHASE(L) + A/1000.                                  LDSYS     54
      LET ITSYS(L) = I                                                   LDSYS     55
      DO TO 45, FOR K = (1)(SITAB)                                       LDSYS     56
      IF SNAME(K) EQ ITSAT(L), GO TO 50                                  LDSYS     57
   45 LOOP                                                               LDSYS     58
C                                                                        LDSYS     59
CCC   ERROR DETECTED                                                     LDSYS     60
C                                                                        LDSYS     61
      LET IRFLG = 1                                                      LDSYS     62
      WRITE ON 6,ITSAT(L),SYNAM(I)                                       LDSYS     63
      FORMAT(S3,* ERROR  SATELLITE =*,A6,*= NOT FOUND, SYSTEM = *,A6)    LDSYS     64
      GO TO 55                                                           LDSYS     65
   50 LET ITSAT(L) = K                                                   LDSYS     66
   55 LOOP                                                               LDSYS     67
   60 LOOP                                                               LDSYS     68
      RETURN                                                             LDSYS     69
      END                                                                LDSYS     70
      SUBROUTINE LOVEH(IRFLG)                                            LOVEH     2
C                                                                        LOVEH     3
C   LOAD VEHICLE DATA                                                    LOVEH     4
```

```
C
      READ FROM 5,NOVEH                                                          LDVEH        5
      FORMAT(I3)                                                                 LDVEH        6
      IF NOVEH LE NVEH, GO TO 1                                                  LDVEH        7
      WRITE ON 6,NOVEH,NVEH                                                      LDVEH        8
      FORMAT(*  ERROR - NUMBER OF VEHICLES INPUT(*,I6,*) EXCEEDS CAPACIT         LDVEH        9
     *Y(*,I6,*)*)                                                               LDVEH       10
      LET IRFLG = 1                                                              LDVEH       11
    1 WRITE ON 6,NOVEH                                                           LDVEH       12
      FORMAT(I8,* VEHICLES INPUT*)                                               LDVEH       13
      WRITE ON 6                                                                 LDVEH       14
      FORMAT(*   NAME        DAYS      ISP       WDV       BOIL     WCONV      LDVEH       15
     * REFT       EXP    LENGTH    NS  SOLID  ID*)                              LDVEH       16
C                                                                                LDVEH       17
CC    LOAD ALL VEHICLE CARDS                                                     LDVEH       18
CC                                                                               LDVEH       19
      DO TO 5, FOR I=(1)(NOVEH)                                                  LDVEH       20
      READ FROM 5,NAMEV(I),DAYSV(I),ISPV(I),WDV(I),WPNUV(I),WCONV(I),            LDVEH       21
     * REFTV(I),EXPV(I),PAYLV(I)                                                LDVEH       22
     *,NSTAG(I),SOLID(I),IDV(I)                                                 LDVEH       23
      FORMAT(A6,8D5.1,2I2,A6)                                                    LDVEH       24
      WRITE ON 6 ,NAMEV(I),DAYSV(I),ISPV(I),WDV(I),WPNUV(I),WCONV(I),            LDVEH       25
     * REFTV(I),EXPV(I),PAYLV(I)                                                LDVEH       26
     *,NSTAG(I),SOLID(I),IDV(I)                                                 LDVEH       27
      FORMAT(S3,A6,8D7.1,2I6,S1,A6)                                              LDVEH       28
      IF NAMEV(I) EQ SEPS, CALL LOSEP(WDV(I),PAYLV(I),WCONV(I),ISPV(I),          LDVEH       29
     * WPNUV(I),EXPV(I),DAYSV(I),REFTV(I))                                      LDVEH       30
    5 LOOP                                                                       LDVEH       31
      RETURN                                                                     LDVEH       32
      END                                                                        LDVEH       33
      SUBROUTINE MARKQ                                                           LDVEH       34
C                                                                                MARKQ        2
CC    MARK ALL PAYLOADS FOR LAUNCH IN ORBIT QUEUE IORB                           MARKQ        3
CC                                                                               MARKQ        4
      LET NQ = 0                                                                 MARKQ        5
      IF ORBQ(IORB) IS EMPTY, RETURN                                             MARKQ        6
      DO TO 5, FOR ALL PAYLD IN ORBQ(IORB)                                       MARKQ        7
      IF LQTIM(PAYLD) GT 3000., RETURN                                           MARKQ        8
      LET NQ = NQ + 1                                                            MARKQ        9
      LET ILOAD(NQ) = PAYLD                                                      MARKQ       10
      IF NQ EQ IL, RETURN                                                        MARKQ       11
    5 LOOP                                                                       MARKQ       12
      RETURN                                                                     MARKQ       13
      END                                                                        MARKQ       14
      SUBROUTINE MCMOD                                                           MARKQ       15
C                                                                                MCMOD        2
CC  STATISTICS FOR MODULES                                                       MCMOD        3
CC                                                                               MCMOD        4
      DO TO 5, FOR I=(1)(MITA3)                                                  MCMOD        5
      IF MDCNT(I) + S121(I) EQ 0, GO TO 1                                        MCMOD        6
      LET S121(I) = S121(I) + MDCNT(I)                                           MCMOD        7
      IF X121(I) LT MDCNT(I), LET X121(I) = MDCNT(I)                             MCMOD        8
      IF N121(I) GT MDCNT(I), LET N121(I) = MDCNT(I)                             MCMOD        9
      IF TRIG NE TRIGS, GO TO 1                                                  MCMOD       10
      IF TRIG EQ 1, GO TO 1                                                      MCMOD       11
      IF N121(I) EQ X121(I), LET N121(I) = 0                                     MCMOD       12
    1 IF NOWAR(I) + S125(I) EQ 0, GO TO 2                                        MCMOD       13
                                                                                 MCMOD       14
```

```
      LET S125(I) = S125(I) + NOWAR(I)                                  MCMOD     15
      IF X125(I) LT NOWAR(I), LET X125(I) = NOWAR(I)                    MCMOD     16
      IF N125(I) GT NOWAR(I), LET N125(I) = NOWAR(I)                    MCMOD     17
      IF TRIG NE TRIGS, GO TO 2                                         MCMOD     18
      IF TRIG EQ 1, GO TO 2                                             MCMOD     19
      IF N125(I) EQ X125(I), LET N125(I) = 0                            MCMOD     20
    2 IF NOFAL(I) + S129(I) EQ 0, GO TO 5                               MCMOD     21
      LET S129(I) = S129(I) + NOFAL(I)                                  MCMOD     22
      IF X129(I) LT NOFAL(I), LET X129(I) = NOFAL(I)                    MCMOD     23
      IF N129(I) GT NOFAL(I), LET N129(I) = NOFAL(I)                    MCMOD     24
      IF TRIG NE TRIGS, GO TO 5                                         MCMOD     25
      IF TRIG EQ 1, GO TO 5                                             MCMOD     26
      IF N129(I) EQ X129(I), LET N129(I) = 0                            MCMOD     27
    5 LOOP                                                              MCMOD     28
      RETURN                                                            MCMOD     29
      END                                                               MCMOD     30
      SUBROUTINE MCSAT                                                  MCSAT      2
C                                                                       MCSAT      3
C   STATISTICS FOR SATELLITES                                           MCSAT      4
C                                                                       MCSAT      5
      DO TO 3, FOR I=(1)(SYORB)                                         MCSAT      6
      IF MOD(I) IS EMPTY, GO TO 3                                       MCSAT      7
      LET S227(I) = S227(I) + SATLF(I)                                  MCSAT      8
      IF X227(I) LT SATLF(I), LET X227(I) = SATLF(I)                    MCSAT      9
      IF N227(I) GT SATLF(I), LET N227(I) = SATLF(I)                    MCSAT     10
      LET A = LFSAT(I)                                                  MCSAT     11
      LET SUMSL(I) = SUMSL(I) + A                                       MCSAT     12
      IF MAXSL(I) LT A, LET MAXSL(I) = A                                MCSAT     13
      IF MINSL(I) GT A, LET MINSL(I) = A                                MCSAT     14
      DO TO 2, FOR ALL MODSY IN MOD(I)                                  MCSAT     15
      LET SUMNU(MODSY) = SUMNU(MODSY) + NUM(MODSY)                      MCSAT     16
      IF MAXNU(MODSY) LT NUM(MODSY), LET MAXNU(MODSY) = NUM(MODSY)      MCSAT     17
      IF MINNU(MODSY) GT NUM(MODSY), LET MINNU(MODSY) = NUM(MODSY)      MCSAT     18
      LET SUMLF(MODSY) = SUMLF(MODSY) + LOADF(MODSY)                    MCSAT     19
      IF MAXLF(MODSY) LT LOADF(MODSY), LET MAXLF(MODSY) = LOADF(MODSY)  MCSAT     20
      IF MINLF(MODSY) GT LOADF(MODSY), LET MINLF(MODSY) = LOADF(MODSY)  MCSAT     21
    2 LOOP                                                              MCSAT     22
      LET A = HALST(I)-BEGST(I)                                         MCSAT     23
      IF A EQ 0., GO TO 3                                               MCSAT     24
      LET P = 100.*SDTST(I)/A                                           MCSAT     25
      LET PERST(I) = PERST(I) + P                                       MCSAT     26
      IF N216(I) GT P, LET N216(I) = P                                  MCSAT     27
      IF X216(I) LT P, LET X216(I) = P                                  MCSAT     28
    3 LOOP                                                              MCSAT     29
      RETURN                                                            MCSAT     30
      END                                                               MCSAT     31
      SUBROUTINE MCVEH                                                  MCVEH      2
C                                                                       MCVEH      3
C   STATISTICS FOR VEHICLES                                             MCVEH      4
C                                                                       MCVEH      5
      DO TO 1, FOR I=(1)(NYEAR)                                         MCVEH      6
      LET SUM39(I) = SUM39(I)+TUGFY(I)                                  MCVEH      7
      IF MAX39(I) LT TUGFY(I), LET MAX39(I) = TUGFY(I)                  MCVEH      8
      IF MIN39(I) GT TUGFY(I),LET MIN39(I) = TUGFY(I)                   MCVEH      9
      LET SUM86(I) = SUM86(I) + SEPFY(I)                                MCVEH     10
      IF MAX86(I) LT SEPFY(I), LET MAX86(I) = SEPFY(I)                  MCVEH     11
      IF MIN86(I) GT SEPFY(I), LET MIN86(I) = SEPFY(I)                  MCVEH     12
```

```
      LET SUM90(I) = SUM90(I) + SUTFY(I)                                MCVEH    13
      IF MIN90(I) GT SUTFY(I), LET MIN90(I) = SUTFY(I)                  MCVEH    14
      IF MAX90(I) LT SUTFY(I), LET MAX90(I) = SUTFY(I)                  MCVEH    15
    1 LOOP                                                              MCVEH    16
      LET IT= 0                                                         MCVEH    17
      LET IT = IT + TUGFY(I),  FOR I=(1)(NYEAR)                         MCVEH    18
      IF MTFLT LT IT, LET MTFLT = IT                                    MCVEH    19
      IF NTFLT GT IT, LET NTFLT = IT                                    MCVEH    20
      LET ITFLT = ITFLT + IT                                            MCVEH    21
      LET IT = 0                                                        MCVEH    22
      LET IT = IT + SUTFY(I), FOR I=(1)(NYEAR)                          MCVEH    23
      LET IFSUT = IFSUT + IT                                            MCVEH    24
      IF MFSUT LT IT, LET MFSUT = IT                                    MCVEH    25
      IF NFSUT GT IT, LET NFSUT = IT                                    MCVEH    26
      LET IT = 0                                                        MCVEH    27
      LET IT = IT + SEPFY(I), FOR I=(1)(NYEAR)                          MCVEH    28
      LET IFSEP = IFSEP + IT                                            MCVEH    29
      IF MFSEP LT IT, LET MFSEP = IT                                    MCVEH    30
      IF NFSEP GT IT, LET NFSEP = IT                                    MCVEH    31
      DO TO 2, FOR I=(1)(3)                                             MCVEH    32
      LET TCVA(I) = TCVA(I) + CVA(I)                                    MCVEH    33
      IF CVA(I) GT XCVA(I), LET XCVA(I) = CVA(I)                        MCVEH    34
      IF CVA(I) LT MCVA(I), LET MCVA(I) = CVA(I)                        MCVEH    35
    2 LOOP                                                              MCVEH    36
      RETURN                                                            MCVEH    37
      END                                                               MCVEH    38
      SUBROUTINE MCSYS                                                  MCSYS     2
C                                                                       MCSYS     3
CCC   STATISTICS FOR SYSTEMS                                            MCSYS     4
C                                                                       MCSYS     5
      DIMENSION SX2(80)                                                 MCSYS     6
      WRITE ON 6,TRIG                                                   MCSYS     7
      FORMAT(*   DISTRIBUTION POINT FOR CYCLE*,I5)                     MCSYS     8
      DO TO 4, FOR I=(1)(STST8)                                         MCSYS     9
      IF SYNAM(I) EQ 0, GO TO 4                                         MCSYS    10
      LET A = 0.                                                        MCSYS    11
      DO TO 6, FOR J=(FSAT(I))(LSAT(I))                                 MCSYS    12
      LET A = A + LFSAT(J)                                              MCSYS    13
    6 LOOP                                                              MCSYS    14
      IF TRIG EQ 1, LET SX2(I) = 0.                                     MCSYS    15
      LET SYLF(I) = SYLF(I) + A                                         MCSYS    16
      IF XSYLF(I) LT A, LET XSYLF(I) = A                                MCSYS    17
      IF NSYLF(I) GT A, LET NSYLF(I) = A                                MCSYS    18
      LET A = HALSY(I) - BEGSY(I)                                       MCSYS    19
      IF A EQ 0., GO TO 4                                               MCSYS    20
      LET P = 100.*SDTSY(I)/A                                           MCSYS    21
      LET PERSY(I) = PERSY(I) + P                                       MCSYS    22
      LET SX2(I) = SX2(I) + P**2                                        MCSYS    23
      LET SIGMA = 0.                                                    MCSYS    24
      LET AN = TRIG                                                     MCSYS    25
      IF TRIG NE 1, LET SIGMA = SQRT((SX2(I)-PERSY(I)**2/AN)/(AN-1.))   MCSYS    26
      LET Q = PERSY(I)/AN                                               MCSYS    27
      IF N200(I) GT P, LET N200(I) = P                                  MCSYS    28
      IF X200(I) LT P, LET X200(I) = P                                  MCSYS    29
      WRITE ON 6,SYNAM(I),A,SDTSY(I),P,Q,SIGMA                          MCSYS    30
      FORMAT(* SYSTEM *,A6,* LIFE *,M5.2.2,* DELAY *,M5.2.2,* AVAIL*    MCSYS    31
     *,D4.6,*AVR AVL *,D4.6,* SIGMA *,D2.6)                             MCSYS    32
```

```
    4 LOOP                                                                  MCSYS      33
      RETURN                                                                MCSYS      34
      END                                                                   MCSYS      35
      ENDOGENOUS EVENT NEWME                                                NEWME       2
C                                                                           NEWME       3
C     REPLACEMENT OR UPGRADING OF ME                                        NEWME       4
C                                                                           NEWME       5
C     FIX UP AND TEST ***************************************************   NEWME       6
        LET IEVME = IEVME + 1                                               NEWME       7
        IF MSTAT(PMOD(NEWME)) EQ UP, CALL SHIP(PSAT(NEWME),PMOD(NEWME))     NEWME       8
        IF MSTAT(PMOD(NEWME)) NE UP, CALL STATUS(IX,IY,5)                   NEWME       9
        DESTROY NEWME                                                       NEWME      10
        RETURN                                                              NEWME      11
        END                                                                 NEWME      12
        ENDOGENOUS EVENT NWSAT                                              NWSAT       2
C                                                                           NWSAT       3
C       THIS ROUTINE WILL ATTEMPT TO SCHEDULE THE LAUNCHING OF A PAYLOAD    NWSAT       4
C                                                                           NWSAT       5
C       ON A VEHICLE.                                                       NWSAT       6
C                                                                           NWSAT       7
C       IT WILL INCLUDE FIRST LAUNCH CHECK TO SET FINAL 6 MONTH LATER GO.   NWSAT       8
C                                                                           NWSAT       9
        LET IEVNW = IEVNW + 1                                               NWSAT      10
        LET IS = PSAT(NWSAT)                                                NWSAT      11
        DESTROY NWSAT                                                       NWSAT      12
        IF TIME GE TIMEG, LET EXMOD = MODS                                  NWSAT      13
        CALL STATUS(IS,0,1)                                                 NWSAT      14
        LET T = TGOSY (ITSYS(IS))                                           NWSAT      15
        IF T EQ 0., GO TO 1                                                 NWSAT      16
        IF TIME GT T, RETURN                                                NWSAT      17
    1 CALL SHIP(IS,0)                                                       NWSAT      18
        LET DELAY = WSATN                                                   NWSAT      19
        IF SSTAT(IS) EQ UP, LET DELAY = WSATU                               NWSAT      20
        IF DELAY GT TIMES - TIME, LET DELAY = TIMES - TIME                  NWSAT      21
        IF DELAY LT 0., LET DELAY = 0.                                      NWSAT      22
        LET DTIME(IS) = TIME + DELAY                                        NWSAT      23
C                                                                           NWSAT      24
C     SCHEDULE MANDATORY LAUNCH                                             NWSAT      25
C                                                                           NWSAT      26
        IF SORTE(ITSAT(IS)) NE 0., RETURN                                   NWSAT      27
        CREATE LAUNC CALLED J                                               NWSAT      28
        LET LQEV(J) = IQ                                                    NWSAT      29
        LET MLEV(IQ) = J                                                    NWSAT      30
        CAUSE LAUNC CALLED J AT TIME + DELAY                                NWSAT      31
        RETURN                                                              NWSAT      32
        END                                                                 NWSAT      33
        SUBROUTINE PASER                                                    PASER       2
C                                                                           PASER       3
C       PHASING ALGORITHM                                                   PASER       4
C                                                                           PASER       5
C       DETERMINE SATELLITE OR RETRIEVED PAYLOAD IN QUEUE                   PASER       6
C                                                                           PASER       7
        LET KSAT = 0                                                        PASER       8
        DO TO 5, FOR J=(1)(NQ)                                              PASER       9
        IF IMOD(ILOAD(J)) + IRT(ILOAD(J)) EQ 0, LET KSAT = 1                PASER      10
    5 LOOP                                                                  PASER      11
C                                                                           PASER      12
```

```
C
C     SORT INTO ORDER OF PHASE ANGLE
C
    6 DO TO 9, FOR K=(1)(NQ-1)
      DO TO 11, FOR J=(K+1)(NQ)
      IF ANGLE(ILOAD(K)) LE ANGLE(ILOAD(J)), GO TO 11
      LET L = ILOAD(K)
      LET ILOAD(K) = ILOAD(J)
      LET ILOAD(J) = L
   11 LOOP
    9 LOOP
CCC
CCC   FIND LARGEST GAP IN CIRCLE
CCC
CCC   OR FLIGHT HAS ONLY MODULES
      LET CX = 0
      LET JSAT = NQ
      DO TO 12, FOR J=(2)(NQ)
      IF ANGLE(ILOAD(J))-ANGLE(   ILOAD(J-1)) LT CX, GO TO 12
      LET CX = ANGLE(ILOAD(J)) - ANGLE(ILOAD(J-1))
      LET JSAT = J
   12 LOOP
      IF 360.-ANGLE(ILOAD(NQ))+ANGLE(ILOAD(1)) GT CX, LET JSAT = 1
      IF JSAT EQ 1, GO TO 14
   13 LET ANGLE(ILOAD(J)) = ANGLE(ILOAD(J)) - 360., FOR J=(JSAT)(NQ)
      GO TO 6
   14 IF KSAT EQ 0, GO TO 50
CCC
CCC   QUIT IF NON-RETRIEVED SATELLITE AT FIRST POSITION
CCC
      DO TO 25, FOR J=(1)(NQ)
      IF IMOD(ILOAD(J))+IRT(ILOAD(J)) EQ 0, GO TO 28
   25 LOOP
      GO TO 50
   28 IF ABS(ANGLE(ILOAD(J))-ANGLE(ILOAD(1))) LT 1., GO TO 50
CCC
CCC   REORDER DELIVERY SEQUENCE
CCC
      IF NQ GT 2, GO TO 21
      LET L = ILOAD(2)
      LET ILOAD(2) = ILOAD(1)
      LET ILOAD(1) = L
      GO TO 50
   21 LET IJ = J
      DO TO 29, FOR K=(IJ)(NQ-1)
      IF ABS(ANGLE(ILOAD(K))-ANGLE(ILOAD(K+1))) GT 1., GO TO 30
      LET J = K + 1
   29 LOOP
   30 IF ANGLE(ILOAD(J))-ANGLE(ILOAD(1)) GT
     *    ANGLE(ILOAD(NQ))-ANGLE(ILOAD(J)), GO TO 22
CCC
CCC   FIND END OF POSITION
CCC
   23 DO TO 26, FOR K=(1)(J/2)
      LET L = ILOAD(J-K+1)
      LET ILOAD(J-K+1) = ILOAD(K)
      LET ILOAD(K) = L
   26 LOOP
```

```
PASER   13
PASER   14
PASER   15
PASER   16
PASER   17
PASER   18
PASER   19
PASER   20
PASER   21
PASER   22
PASER   23
PASER   24
PASER   25
PASER   26
PASER   27
PASER   28
PASER   29
PASER   30
PASER   31
PASER   32
PASER   33
PASER   34
PASER   35
PASER   36
PASER   37
PASER   38
PASER   39
PASER   40
PASER   41
PASER   42
PASER   43
PASER   44
PASER   45
PASER   46
PASER   47
PASER   48
PASER   49
PASER   50
PASER   51
PASER   52
PASER   53
PASER   54
PASER   55
PASER   56
PASER   57
PASER   58
PASER   59
PASER   60
PASER   61
PASER   62
PASER   63
PASER   64
PASER   65
PASER   66
PASER   67
PASER   68
PASER   69
```

```
      GO TO 50
   22 IF J EQ NQ, GO TO 23
C
C     FIND END OF POSITION
C
      LET IJ =(NQ-J+1)/2
      DO TO 27, FOR K=(J)(J+IJ-1)
      LET L = ILOAD(NQ-K+J)
      LET ILOAD(NQ-K+J) = ILOAD(K)
      LET ILOAD(K) = L
   27 LOOP
      LET J = NQ
      GO TO 23
C
C     PHASING SETUP COMPLETE
C
   50 RETURN
      END
      SUBROUTINE PAYLQ(IS,IM,ILL)
C
C  ENTER PAYLOAD INTO LOADING QUEUE AND ORBIT QUEUE
C
      CALL QDMP(IS,IM,ILL)
      IF IM EQ 0, GO TO 1
      IF TIME + DELTA GT LIMIT, GO TO 2
      IF EXMOD NE 100, GO TO 1
    2           LET ILL = 1
      RETURN
    1 LET IQ = 0
      IF ILL NE 0, RETURN
      CREATE PAYLD CALLED IX
      LET ISAT(IX) = IS
      LET IMOD(IX) = IM
      IF IM NE 0, GO TO 5
      LET XSAT(IS)=EXMOD
      IF EXMOD EQ 100, LET PAYWT(IX)=EXWT(ITSAT(IS))
      IF EXMOD NE 100, LET PAYWT(IX)=SWT(ITSAT(IS))
      GO TO 10
    5 LET PAYWT(IX)=MODWT(NOMOD(IM))
   10 LET ANGLE(IX)=PHASE(IS)
      LET IRT(IX) = RTFLG
      LET GOTIM(IX) = 0.
      IF IM NE 0, LET PAYLN(IX) = 0.
      IF IM EQ 0, LET PAYLN(IX) = SVOL(ITSAT(IS))
      CALL REDUN(IS,IM)
      IF DELTA LT 0., LET DELTA = 0.
      LET LQTIM(IX) = TIME + DELTA + PRIOR(ITSAT(IS))
      LET IQ = IX
      LET MLEV(IX) = 0
      IF ORBQ(IORB) IS EMPTY, GO TO 15
      IF LQTIM(LORBQ(IORB)) GT LQTIM(IX), GO TO 15
      LET SORBQ(LORBQ(IORB)) = IX
      LET PORBQ(IX) = LORBQ(IORB)
      LET SORBQ(IX) = 0
      LET LORBQ(IORB) = IX
      RETURN
   15 FILE IX IN ORBQ(IORB)
```

```
PASER   70
PASER   71
PASER   72
PASER   73
PASER   74
PASER   75
PASER   76
PASER   77
PASER   78
PASER   79
PASER   80
PASER   81
PASER   82
PASER   83
PASER   84
PASER   85
PASER   86
PASER   87
PAYLQ    2
PAYLQ    3
PAYLQ    4
PAYLQ    5
PAYLQ    6
PAYLQ    7
PAYLQ    8
PAYLQ    9
PAYLQ   10
PAYLQ   11
PAYLQ   12
PAYLQ   13
PAYLQ   14
PAYLQ   15
PAYLQ   16
PAYLQ   17
PAYLQ   18
PAYLQ   19
PAYLQ   20
PAYLQ   21
PAYLQ   22
PAYLQ   23
PAYLQ   24
PAYLQ   25
PAYLQ   26
PAYLQ   27
PAYLQ   28
PAYLQ   29
PAYLQ   30
PAYLQ   31
PAYLQ   32
PAYLQ   33
PAYLQ   34
PAYLQ   35
PAYLQ   36
PAYLQ   37
PAYLQ   38
PAYLQ   39
PAYLQ   40
```

```
      RETURN                                                                    PAYLQ      41
      END                                                                       PAYLQ      42
      SUBROUTINE PROP(MARKP)                                                    PROP        2
      DIMENSION A(20)                                                           PROP        3
C                                                                               PROP        4
CC    DETECT AND COUNT SORTIES AND MODULES IN LOADING QUEUE                     PROP        5
C                                                                               PROP        6
      LET KX = 0                                                                PROP        7
      LET NMD = 0                                                               PROP        8
      DO TO 10, FOR J=(1)(NQ)                                                   PROP        9
      IF SORTE(ITSAT(ISAT(ILOAD(J)))) NE 0, LET KX = J                          PROP       10
      IF IMOD(ILOAD(J)) NE 0, LET NMD = NMD + 1                                 PROP       11
   10 LOOP                                                                      PROP       12
      IF KX GT 1, GO TO 70                                                      PROP       13
      IF KX EQ 1, GO TO 90                                                      PROP       14
C                                                                               PROP       15
CC    VOLUME(LENGTH) CONSTRAINT IS CHECKED                                      PROP       16
C                                                                               PROP       17
      LET PALEN = PAYLV(RQSUT(IORB))                                            PROP       18
      IF RQUP(IORB) NE 0, LET PALEN = PAYLV(RQUP(IORB))                         PROP       19
      LET  SU =(NMD+NINSU-1)/NINSU                                              PROP       20
      LET PAY = SU*LENSU                                                        PROP       21
      DO TO 20, FOR L=(1)(NQ)                                                   PROP       22
      IF IMOD(ILOAD(L))+IRT(ILOAD(L)) EQ 0,                                     PROP       23
     *   LET PAY = PAY + SVOL(ITSAT(ISAT(ILOAD(L))))                            PROP       24
C****  CHECK DOWN LENGTH                                                        PROP       25
   20 LOOP                                                                      PROP       26
      IF PAY GT PALEN, GO TO 70                                                 PROP       27
C                                                                               PROP       28
CC    CALL PERFORMANCE COMPUTATION ROUTINE                                      PROP       29
C                                                                               PROP       30
      LET XX = PALEN - PAY                                                      PROP       31
      CALL PROP2(MARKP,XX)                                                      PROP       32
      RETURN                                                                    PROP       33
C                                                                               PROP       34
CC    PAYLOAD COMBINATION IS REJECTED - PERFORMANCE, LENGTH OR SORTIES          PROP       35
C                                                                               PROP       36
   70 LET W(IORB) = -10.                                                        PROP       37
      RETURN                                                                    PROP       38
C                                                                               PROP       39
CC   SINGLE SORTIE OPTION                                                       PROP       40
C                                                                               PROP       41
   90 LET W(IORB) = -50.                                                        PROP       42
      LET NL(IORB) = 1                                                          PROP       43
      LET GOTIM(ILOAD(1))= 6./8640.                                             PROP       44
      LET ORBTM(IORB) = SORTE(ITSAT(ISAT(ILOAD(1))))                            PROP       45
      LET ANMD(IORB) = 0                                                        PROP       46
      LET PQUE(IORB) = ILOAD(1)                                                 PROP       47
      RETURN                                                                    PROP       48
      END                                                                       PROP       49
      SUBROUTINE PROP2(MARKP,PAY)                                               PROP2       2
C                                                                               PROP2       3
CC    COMPUTE PROPELLANT REQUIRED TO DELIVER NQ ITEMS IN CLOAD ARRAY            PROP2       4
CC                                                                              PROP2       5
C                                                                               PROP2       6
      DIMENSION PLEG(20),DVLEG(20),THETA(20),A(20)                              PROP2       7
      DIMENSION BOIL(20)                                                        PROP2       8
```

```
CCC     GET VEHICLE DATA                                                        PROP2      9
C                                                                               PROP2     10
        IF RQSEP(IORB) EQ 0, GO TO 1                                            PROP2     11
        LET I = RQSEP(IORB)                                                     PROP2     12
        LET FS = WDV(I) + REFTV(I)                                              PROP2     13
        LET FD = WDV(I)                                                         PROP2     14
    1   LET DUMMY= 0                                                            PROP2     15
        LET JK = RQUP(IORB)                                                     PROP2     16
        IF JK EQ 0, LET JK = 1                                                  PROP2     17
        LET DAYS = DAYSV(JK)                                                    PROP2     18
        LET WCONS = WCONV(RQSUT(IORB))                                          PROP2     19
        LET DV = ORBDV(IORB)                                                    PROP2     20
        LET RA = ORBRA(IORB)                                                    PROP2     21
        LET VCO = 25936.                                                        PROP2     22
        LET P1 = ORBPD(IORB)                                                    PROP2     23
        LET WRET = 0.                                                           PROP2     24
        LET WDEP = 0.                                                           PROP2     25
        LET WSERV = 0.                                                          PROP2     26
        DO TO 5, FOR J=(1)(NQ)                                                  PROP2     27
        CALL QUAD(ANGLE(ILOAD(J)))                                              PROP2     28
        IF IMOD(ILOAD(J)) EQ 0, LET WDEP = WDEP + PAYWT(ILOAD(J))               PROP2     29
        IF IMOD(ILOAD(J)) NE 0, LET WSERV = WSERV + PAYWT(ILOAD(J))             PROP2     30
        IF IRT(ILOAD(J)) EQ 0, GO TO 5                                          PROP2     31
        LET WDEP = WDEP - PAYWT(ILOAD(J))                                       PROP2     32
        LET WRET = WRET + PAYWT(ILOAD(J))                                       PROP2     33
    5 LOOP                                                                      PROP2     34
                                                                                PROP2     35
CCC  COMPUTE PERFORMANCE - UP/DOWN PAYLOADS                                     PROP2     36
C                                                                               PROP2     37
        LET WSERV = WSERV + WTSU* SU                                            PROP2     38
        LET WUPL = WDEP + WSERV                                                 PROP2     39
        LET WSPL = WRET + WSERV                                                 PROP2     40
        IF PSERV EQ 1, LET WSPL = WRET                                          PROP2     41
        IF PSERV EQ 2, LET WSPL = WRET + WTSU * SU                              PROP2     42
        IF RQUP(IORB) EQ 0, GO TO 100                                           PROP2     43
        LET WBOIL = WPNUV(JK)                                                   PROP2     44
        LET NS = NSTAG(JK)                                                      PROP2     45
        IF NS EQ 0, LET NS = 1                                                  PROP2     46
        DO TO 40, FOR NK=(1)(NS)                                                PROP2     47
        LET JX = JK + NK - 1                                                    PROP2     48
        IF EXVEH EQ 0, LET EXVEH = EXPV(JX)                                     PROP2     49
        LET XVEH = EXVEH                                                        PROP2     50
        CALL LINKT(NK,ISPV(JX),WDV(JX),WPNUV(JX),WCONV(JX),XVEH,                PROP2     51
       * SOLID(JX),WCONV(RQSUT(IORB)),TRIN)                                     PROP2     52
   40 LOOP                                                                      PROP2     53
        IF NS GT 1, CALL TWOBR(DV,DV1(IORB))                                    PROP2     54
    6   LET NLEG = 1                                                            PROP2     55
        LET PLEG(1) = WUPL                                                      PROP2     56
        LET DVLEG(1) = DV                                                       PROP2     57
        LET BOIL(1) = WBOIL*6.                                                  PROP2     58
        LET MARKP = 0                                                           PROP2     59
        IF NQ EQ 1, GO TO 1000                                                  PROP2     60
        LET GDAY = DAYS - .5                                                    PROP2     61
C                                                                               PROP2     62
CCC  COMPUTE PROPELLANT FOR SERVICING                                           PROP2     63
C                                                                               PROP2     64
                                                                                PROP2     65
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      CALL PASER                                                        PROP2      66
   50 LET PANGL(1) = 0.                                                 PROP2      67
      LET PANGL(J) = ANGLE(ILOAD(J)) - ANGLE(ILOAD(J-1)),               PROP2      68
     * FOR J=(2)(NQ)                                                   PROP2      69
      LET TO = 0.                                                       PROP2      70
      LET TO = TO + ABS(PANGL(J)), FOR J=(2)(NQ)                        PROP2      71
      DO TO 60, FOR MFLT = (2)(NQ)                                      PROP2      72
      LET X = WSERV                                                     PROP2      73
      LET NFF = MFLT                                                    PROP2      74
      DO TO 54, FOR J=(1)(NFF-1)                                        PROP2      75
      IF IRT(ILOAD(J)) NE 0, LET X = X + PAYWT(ILOAD(J))                PROP2      76
   54 LOOP                                                              PROP2      77
      DO TO 55, FOR J=(NFF)(NQ)                                         PROP2      78
      IF IMOD(ILOAD(J)) EQ 0, LET X = X + PAYWT(ILOAD(J))               PROP2      79
   55 LOOP                                                              PROP2      80
C                                                                       PROP2      81
CCC   COMPUTE PHASING PROPELLANT                                        PROP2      82
C                                                                       PROP2      83
      LET FLTIM(NFF) = 0.                                               PROP2      84
      IF PANGL(NFF) EQ 0., GO TO 60                                     PROP2      85
      IF ABS(PANGL(NFF)) LT 1., GO TO 60                                PROP2      86
      LET IETA = ABS(PANGL(NFF))/TO*GDAY*24./P1 + .2                    PROP2      87
      IF IETA LE 0, LET IETA = 1                                        PROP2      88
      LET ETA = IETA                                                    PROP2      89
      LET MARKP = 1                                                     PROP2      90
      LET PO = P1*(1.-PANGL(NFF)/(360.*ETA))                            PROP2      91
      LET TO = TO - ABS(PANGL(NFF))                                     PROP2      92
      LET FLTIM(NFF) = PO*ETA/(24.*30.*12.)                             PROP2      93
      LET GDAY = GDAY - PO/24.*ETA                                      PROP2      94
      IF GDAY LT -.5, GO TO 70                                          PROP2      95
      IF PO LT .3535*P1, GO TO 70                                       PROP2      96
      LET RP = RA*(2.*(PO/P1)**(2./3.)-1.)                              PROP2      97
      LET VCP = VCO * SQRT(RO/RP)                                       PROP2      98
      LET DVO = 2.*VCP*(SQRT(1./(RA/RP))-SQRT(2./((RA/RP)*(1.+RA/RP)))) PROP2      99
      LET DVO = ABS(DVO)                                                PROP2     100
      LET NLEG = NLEG + 1                                               PROP2     101
      LET PLEG(NLEG) = X                                                PROP2     102
      LET DVLEG(NLEG) = DVO                                             PROP2     103
      LET BOIL(NLEG) = WBOIL*PO*ETA                                     PROP2     104
      LET THETA(NLEG-1) = PANGL(NFF)                                    PROP2     105
   60 LOOP                                                              PROP2     106
 1000 LET NLEG = NLEG + 1                                               PROP2     107
      LET PLEG(NLEG) = WSPL                                             PROP2     108
      LET DVLEG(NLEG) = DV                                              PROP2     109
      LET BOIL(NLEG) = WBOIL*6.                                         PROP2     110
      IF RQSEP(IORB) NE 0, GO TO 64                                     PROP2     111
   63 IF EXVEH EQ 0, GO TO 670                                          PROP2     112
      IF WRET NE 0., GO TO 70                                           PROP2     113
       LET NLEG = NLEG - 1                                              PROP2     114
  670 LET DUMMY = 0                                                     PROP2     115
C                                                                       PROP2     116
CC    OBTAIN PROPELLANT REQUIREMENTS FOR TUG TYPE VEHICLES              PROP2     117
C                                                                       PROP2     118
      LET JKO = 0                                                       PROP2     119
      CALL CONEC(NS,JK,JKO)                                             PROP2     120
      CALL PRFORM(DVLEG,PLEG,BOIL,NLEG,WP,NM)                           PROP2     121
      IF WP LT 0., GO TO 65                                             PROP2     122
```

```
      LET FLTIM(1) = 6./3640.
      LET FLTIM(J) = FLTIM(J-1) + 6./3640., FOR J=(2)(NQ)
      LET FLY = FLTIM(NQ) + 6./3640.
      GO TO 65

      SEPS PERFORMANCE COMPUTATIONS

64    IF ISEPS EQ 0, GO TO 63
      IF EXVEH NE 0, GO TO 70
      CALL SEPSV(NLEG-2,P1,VCO,THETA(1),PLEG(2))
      LET PLEG(NLEG) = SWON(ISEPS)
      LET JKO = ISEPS
      CALL CONEC(NS,CHEM,ISEPS)
      IF MSEP(ISEPS) EQ 1, LET NEXIT(ISEPS) = LEXIT(ISEPS)
      LET LEXIT(ISEPS) = NEXIT(ISEPS)
      CALL PRFORM(DVLEG,PLEG,BOIL,NLEG,WP,**NEXIT(ISEPS),**MSEP(ISEPS))
      LET MSEP(ISEPS) = 1
      LET WP = 10
      LET WUSEP = 0
      LET WDNSP = 0

        ------ SEPS OPTIONS   ----------
      NEXIT VALUES

        1     SEPS UP NEW AT MIN ALT - SET WUSEP AND LENGTH AND WEIGHT CHEC
        2     SEPS UP AT SYNC EQ - DOES PHASING ONLY - SAME AS ABOVE
        3     NO GOOD
        4     NO GOOD
        5     OK - SEPS DOWN TO MEET TUG
        6     OK - THEY MEET AT SYNC EQ
        7     NO GOOD
        8     NO GOOD
        9     NO GOOD
        10    SEPS BROUGHT DOWN - NO UP PAYLOADS

      GO TO (200,210,110,110,220,230,110,235,250,240),NEXIT(ISEPS)
0     LET DUMMY = 0
      IF LEXIT(ISEPS) NE 0, GO TO 250
      IF LSEP GT PAY, GO TO 70
      IF SCOOT EQ 0, GO TO 201
      IF NQ GT 1, GO TO 70
1     LET WUSEP = FS
      GO TO 260
0     LET WUSEP = FS
      IF LEXIT(ISEPS) NE 0, GO TO 250
      IF LSEP GT PAY, GO TO 70
      GO TO 260
0     LET DUMMY = 0
```

PROP2 123 … PROP2 179

```
    IF SCOOT GT 0, GO TO 70                                                    PROP2   180
    GO TO 260                                                                  PROP2   181
30  LET DUMMY = 0                                                              PROP2   182
    GO TO 260                                                                  PROP2   183
35  LET DUMMY= 0                                                               PROP2   184
    LET NQ = -2                                                                PROP2   185
    LET NMD = 0                                                                PROP2   186
    LET WONSP=FD                                                               PROP2   187
     GO TO 260                                                                 PROP2   188
40  LET DUMMY = 0                                                              PROP2   189
50  LET DUMMY = 0                                                              PROP2   190
    LET NQ = -1                                                                PROP2   191
    LET NMD = 0                                                                PROP2   192
    LET WONSP= FD                                                              PROP2   193
    LET WP = -10.                                                              PROP2   194
    GO TO 260                                                                  PROP2   195
60  LET DUMMY =0                                                               PROP2   196
    LET A(I) = 0., FOR I=(1)(20)                                               PROP2   197
    CALL TPHAS(A,NLEG)                                                         PROP2   198
    LET TUP = A(1)                                                             PROP2   199
    LET TDOWN = A(NLEG)                                                        PROP2   200
    LET TDOWN = TDOWN - TIME + AVSEP(ISEPS) - PADT                             PROP2   201
    IF TDOWN LT 0., LET TDOWN = 0.                                             PROP2   202
    LET FLTIM(1) = TUP + 6./8640.                                              PROP2   203
    LET M = 2                                                                  PROP2   204
    DO TO 66, FOR I=(2)(NQ)                                                    PROP2   205
    LET FLTIM(I) = FLTIM(I-1)                                                  PROP2   206
    IF ABS(PANGL(I)) LT 1., GO TO 66                                           PROP2   207
    LET FLTIM(I) = A(M) + FLTIM(1)                                             PROP2   208
    LET M = M + 1                                                              PROP2   209
66  LOOP                                                                       PROP2   210
    LET FLY = 0.                                                               PROP2   211
    IF NQ GT 0, LET FLY = FLTIM(NQ) + 1./3640.                                 PROP2   212
    IF NQ LT 0, GO TO 65                                                       PROP2   213
    IF FLY + TDOWN GT TLIMS, GO TO 70                                          PROP2   214
65  LET W(IORB) = WP                                                           PROP2   215
    IF NQ LT 3, GO TO 67                                                       PROP2   216
    IF W(IORB) LT 0., RETURN                                                   PROP2   217
67  LET DUMMY = 0                                                              PROP2   218
                                                                               PROP2   219
SAVE PREVIOUS GOOD LAUNCH SETUP FOR NEXT FLIGHT (IF SEQUENCE ENDS )            PROP2   220
                                                                               PROP2   221
    LET NL(IORB) = NQ                                                          PROP2   222
    LET GOTIM(ILOAD(J)) = FLTIM(J), FOR J=(1)(NQ)                              PROP2   223
    LET ORBTM(IORB) = FLY                                                      PROP2   224
    LET ANMD(IORB) = NMD                                                       PROP2   225
    LET CITEM(ILOAD(J)) = ILOAD(J+1), FOR J=(1)(NQ-1)                          PROP2   226
    LET PQUE(IORB) = ILOAD(1)                                                  PROP2   227
    RETURN                                                                     PROP2   228
                                                                               PROP2   229
    PAYLOAD COMBINATION IS REJECTED - PERFORMANCE, LENGTH OR SORTIES           PROP2   230
                                                                               PROP2   231
70  LET W(IORB) = -10.                                                         PROP2   232
    LET WUSEP = 0                                                              PROP2   233
    LET WONSP = 0                                                              PROP2   234
    RETURN                                                                     PROP2   235
                                                                               PROP2   236
```

```
C    SHUTTLE ONLY OPTION                                                    PROP2    237
C                                                                           PROP2    238
  100 IF WUPL GT WCONS, GO TO 70                                            PROP2    239
      IF WSPL GT WCONS, GO TO 70                                            PROP2    240
      LET NL(IORB) = NQ                                                     PROP2    241
      LET W(IORB) = 100.*(1.-WUPL/WCONS)                                    PROP2    242
      LET GOTIM(ILOAD(J)) = 6./8640., FOR J=(1)(NQ)                         PROP2    243
      LET ANMD(IORB) = SU                                                   PROP2    244
      LET ORBTM(IORB) = 24./8640.                                           PROP2    245
      LET CITEM(ILOAD(J)) = ILOAD(J+1), FOR J=(1)(NQ-1)                     PROP2    246
      LET PQUE(IORB) = ILOAD(1)                                             PROP2    247
      RETURN                                                                PROP2    248
  110 LET X= 0                                                              PROP2    249
      GO TO 70                                                              PROP2    250
      END                                                                   PROP2    251
      SUBROUTINE QDMP(IS,IM,ILL)                                            QDMP       2
C                                                                           QDMP       3
CC    REMOVES EARLIER DUPLICATE PAYLOAD FROM LOADING QUEUE                  QDMP       4
CC    ALSO BLOCKS MODULES FROM ENTERING QUEUE                               QDMP       5
C                                                                           QDMP       6
      LET IORB = ORBIT(ITSAT(IS))                                           QDMP       7
      LET ILL = 0                                                           QDMP       8
      IF SORTE(ITSAT(IS)) NE 0., RETURN                                     QDMP       9
      IF RTFLG EQ 0, GO TO 1                                                QDMP      10
      IF NPOS(IS) GT 1, RETURN                                              QDMP      11
    1 IF ORBQ(IORB) IS EMPTY, GO TO 3                                       QDMP      12
C                                                                           QDMP      13
CCC   LOGIC FOR SATELLITE ENTERING QUEUE AND FLUSHING ALL MODULES FROM      QDMP      14
CCC   PREVIOUS SATELLITE FROM QUEUE                                         QDMP      15
C                                                                           QDMP      16
      IF IM NE 0, GO TO 8                                                   QDMP      17
      DO TO 5, FOR ALL PAYLD IN ORBQ(IORB)                                  QDMP      18
      IF ISAT(PAYLD) NE IS, GO TO 5                                         QDMP      19
      IF IRT(PAYLD) NE 0, GO TO 5                                           QDMP      20
      IF IMOD(PAYLD) EQ 0, GO TO 7                                          QDMP      21
      CALL DROPQ(PAYLD,IORB)                                                QDMP      22
      LET NL(IORB) = 0                                                      QDMP      23
    5 LOOP                                                                  QDMP      24
    3 RETURN                                                                QDMP      25
    7 LET ILL = 1                                                           QDMP      26
      RETURN                                                                QDMP      27
C                                                                           QDMP      28
CCC   LOGIC FOR MODULES ENTERING QUEUE - SATELLITE ALREADY IN QUEUE         QDMP      29
CCC   CAN INHIBIT MODULE ENTRY                                              QDMP      30
C                                                                           QDMP      31
    8 DO TO 9, FOR ALL PAYLD IN ORBQ(IORB)                                  QDMP      32
      IF ISAT(PAYLD) NE IS, GO TO 9                                         QDMP      33
      IF IMOD(PAYLD) EQ 0, GO TO 7                                          QDMP      34
      IF IMOD(PAYLD) NE IM, GO TO 9                                         QDMP      35
      CALL DROPQ(PAYLD,IORB)                                                QDMP      36
      LET NL(IORB) = 0                                                      QDMP      37
      RETURN                                                                QDMP      38
    9 LOOP                                                                  QDMP      39
      RETURN                                                                QDMP      40
      END                                                                   QDMP      41
      ENDOGENOUS EVENT QWAIT                                                QWAIT      2
C                                                                           QWAIT      3
```

```
C
C    MODULES WAIT ONE WEEK BEFORE ENTERING LOADING QUEUE
C
     LET IEVQW = IEVQW + 1
     LET IS = PSAT(QWAIT)
     LET IM = PMOD(QWAIT)
     LET DELAY = TIMEA(QWAIT)
     DESTROY QWAIT
     IF TIME + DELAY GT TGO(IS), RETURN
     CALL REDUN(IS,IM)
     IF DELTA GT 0., CALL PAYLQ(IS,IM,ILL)
     IF DELTA GT 0., RETURN
     CALL SHIP(IS,IM)
     IF IQ EQ 0,RETURN
     CREATE LAUNC CALLED K
     LET LQEV(K) = IQ
     LET MLEV(IQ) = K
     CAUSE LAUNC CALLED K AT TIME + DELAY
     RETURN
     END
     SUBROUTINE QUAD(A)
   5 IF A GT 0., GO TO 10
     LET A = A + 360.
     GO TO 5
  10 IF A LT 360., RETURN
     LET A = A - 360.
     GO TO 10
     END
     SUBROUTINE REDUN(IS,IM)
     LET DELTA = 0
     IF IM EQ 0, RETURN
     IF MSTAT(IM) EQ 3, LET EDO(IM) = 1
C
C    FIND REDUNDANT SUBSYSTEM
C
     DO TO 5, FOR ALL MODSY IN MOD(IS)
     LET IX = NRU(MODSY)
     IF IX EQ 0, GO TO 4
     IF IX EQ 100, GO TO 4
     IF IX EQ 1, GO TO 3
     LET IB = 0
     LET IY = 0
     LET IK = MODSY
C
C    DETERMINE IF SUBSYSTEM CONTAINS THIS ELEMENT AND COUNT FAILURES
C
     DO TO 1, FOR I=(1)(IX)
     IF IM EQ IK, LET IY = 1
     IF EDO(IK) NE 0, LET IB = IB + 1
     LET IN = IK
     LET IK = SMOD(IK)
   1 LOOP
     IF IY NE 0, GO TO 6
     LET MODSY = IN
     GO TO 5
C
C    SINGLE FREEBIE FOUND
C
```

```
QWAIT     4
QWAIT     5
QWAIT     6
QWAIT     7
QWAIT     8
QWAIT     9
QWAIT    10
QWAIT    11
QWAIT    12
QWAIT    13
QWAIT    14
QWAIT    15
QWAIT    16
QWAIT    17
QWAIT    18
QWAIT    19
QWAIT    20
QWAIT    21
QWAIT    22
QUAD      2
QUAD      3
QUAD      4
QUAD      5
QUAD      6
QUAD      7
QUAD      8
QUAD      9
REDUN     2
REDUN     3
REDUN     4
REDUN     5
REDUN     6
REDUN     7
REDUN     8
REDUN     9
REDUN    10
REDUN    11
REDUN    12
REDUN    13
REDUN    14
REDUN    15
REDUN    16
REDUN    17
REDUN    18
REDUN    19
REDUN    20
REDUN    21
REDUN    22
REDUN    23
REDUN    24
REDUN    25
REDUN    26
REDUN    27
REDUN    28
REDUN    29
REDUN    30
REDUN    31
```

```
    3 IF IM NE MODSY, GO TO 5                                                  REDUN      32
      LET DELTA = 3000.                                                        REDUN      33
      RETURN                                                                   REDUN      34
C                                                                              REDUN      35
CCC   QUICK EXIT ON NRU OR SINGLE STRAND                                       REDUN      36
C                                                                              REDUN      37
    4 IF IM EQ MODSY, RETURN                                                   REDUN      38
    5 LOOP                                                                     REDUN      39
      RETURN                                                                   REDUN      40
C                                                                              REDUN      41
CCC   DETERMINE IF ELEMENT IS A FREEBIE                                        REDUN      42
C                                                                              REDUN      43
    6 LET IB = IX - NRU(SMOD(MODSY)) - 18                                      REDUN      44
      LET A = IB                                                               REDUN      45
      IF IB GE 0, LET DELTA = 3000. + A*1000.                                  REDUN      46
      LET IA = 0                                                               REDUN      47
      IF IX GT 2, LET IA= NRU(SMOD(SMOD(MODSY)))                               REDUN      48
      IF IA EQ 0, RETURN                                                       REDUN      49
      IF IB LT 0, RETURN                                                       REDUN      50
      IF IB LT IA, LET DELTA = -3000.                                          REDUN      51
      RETURN                                                                   REDUN      52
      END                                                                      REDUN      53
      ENDOGENOUS EVENT REFMO                                                   REFMO       2
C                                                                              REFMO       3
CCC   THIS ROUTINE TAKES CARE OF REFURB OF MODULES                             REFMO       4
C                                                                              REFMO       5
      LET IEVMO = IEVMO + 1                                                    REFMO       6
      LET IM = PMOD (REFMO)                                                    REFMO       7
      LET MDCNT(IM) = MDCNT(IM) + 1                                            REFMO       8
      DESTROY REFMO                                                            REFMO       9
      IF TRIG EQ 0, WRITE ON 6,TIME,VNAME(IM)                                  REFMO      10
      FORMAT(S5,M5.2.2,S43,A6,S4,*REFURBISHED*)                                REFMO      11
      RETURN                                                                   REFMO      12
      END                                                                      REFMO      13
      ENDOGENOUS EVENT REFSA                                                   REFSA       2
C                                                                              REFSA       3
CCC   THIS ROUTINE TAKES CARE OF REFURB OF SATELLITES                          REFSA       4
C                                                                              REFSA       5
      LET IEVSA = IEVSA + 1                                                    REFSA       6
      RETURN                                                                   REFSA       7
      END                                                                      REFSA       8
      ENDOGENOUS EVENT REFVE                                                   REFVE       2
C                                                                              REFVE       3
CCC   THIS ROUTINE TAKES CARES OF REFURB OF VEHICLES                           REFVE       4
C                                                                              REFVE       5
      LET IEVVE = IEVVE + 1                                                    REFVE       6
      IF TRIG NE 0, GO TO 2                                                    REFVE       7
      LET TE = TIME                                                            REFVE       8
      LET I = DPART (TE)                                                       REFVE       9
      LET J = HPART (TE) + 1                                                   REFVE      10
      LET K = MPART (TE) + 1                                                   REFVE      11
      WRITE ON 6,I,J,K,VNAME (REFVE),PMOD (REFVE)                              REFVE      12
      FORMAT(*0     *,I5,*.*,I2,*.*,I2,S63,A6,I3,S1,*AVAILABLE*)               REFVE      13
    2 LET IC = 0                                                               REFVE      14
      IF VNAME(REFVE) EQ SEPS, GO TO 6                                         REFVE      15
      IF VNAME(REFVE) EQ SHUT, GO TO 5                                         REFVE      16
      IF VNAME(REFVE) EQ KPAD, GO TO 8                                         REFVE      17
```

```
C
C    UPPER STAGE COMPLETES REFURBISHMENT                                    REFVE   18
C                                                                           REFVE   19
                                                                            REFVE   20
      DO TO 1, FOR I=(1)(NTUG)                                              REFVE   21
      IF VTUG(I) GT 0, LET IC = IC + 1                                      REFVE   22
    1 LOOP                                                                  REFVE   23
      LET VTUG(PMOD(REFVE)) = 1                                             REFVE   24
      IF IC NE 0, GO TO 15                                                  REFVE   25
      GO TO 10                                                              REFVE   26
C                                                                           REFVE   27
CCC   SHUTTLE COMPLETES REFURBISHMENT                                       REFVE   28
C                                                                           REFVE   29
    5 LET IC = IC + VSHUT(I), FOR I=(1)(NSHUT)                              REFVE   30
      LET VSHUT(PMOD(REFVE)) = 1                                            REFVE   31
      IF IC NE 0, GO TO 15                                                  REFVE   32
      GO TO 10                                                              REFVE   33
C                                                                           REFVE   34
CCC   SEPS COMPLETES REFURBISHMENT                                          REFVE   35
C                                                                           REFVE   36
    6 LET IC = IC + VSEPS(I), FOR I=(1)(NSEPS)                              REFVE   37
      LET VSEPS(PMOD(REFVE)) = 1                                            REFVE   38
      LET AVSEP(PMOD(REFVE)) = TIME                                         REFVE   39
      IF IC NE 0, GO TO 15                                                  REFVE   40
      GO TO 10                                                              REFVE   41
C                                                                           REFVE   42
CCC   PAD COMPLETES REFURBISHMENT                                           REFVE   43
C                                                                           REFVE   44
    8 LET J = PSAT(REFVE)                                                   REFVE   45
      LET IC = IC + VPAD(I), FOR I=(NPAD1(J))(NPAD2(J))                     REFVE   46
      LET VPAD(PMOD(REFVE)) = 1                                             REFVE   47
      IF IC NE 0, GO TO 15                                                  REFVE   48
C                                                                           REFVE   49
CCC   FORCE ANY FLIGHT FOUND LEFT IN THE LOADING QUEUE DUE TO LACK OF VE    REFVE   50
C                                                                           REFVE   51
   10 DO TO 11, FOR I=(1)(NORBS)                                            REFVE   52
      LET IORB = I                                                          REFVE   53
      IF ORBQ(IORB) IS EMPTY, GO TO 11                                      REFVE   54
      IF VNAME(REFVE) NE SEPS, GO TO 12                                     REFVE   55
      IF RQSEP(IORB) EQ 0, GO TO 12                                         REFVE   56
      LET EXORB(IORB) = 0                                                   REFVE   57
      LET W(IORB) = -10.                                                    REFVE   58
      LET NL(IORB) = 0                                                      REFVE   59
12    LET DUMMY = 0                                                         REFVE   60
      IF W(IORB) GE 0., GO TO 11                                            REFVE   61
      CALL GETV(IGO)                                                        REFVE   62
      IF IGO NE 0, GO TO 11                                                 REFVE   63
      CALL SHIP(-1,0)                                                       REFVE   64
   11 LOOP                                                                  REFVE   65
   15 DESTROY REFVE                                                         REFVE   66
      RETURN                                                                REFVE   67
UKPAD PAD                                                                   REFVE   68
      END                                                                   REFVE   69
      ENDOGENOUS EVENT REMOV                                                REMOV    2
C                                                                           REMOV    3
CCC   NOTE STATUS OF REMOVAL OF SATELLITE FROM ORBIT                        REMOV    4
C                                                                           REMOV    5
      LET IEVOV = IEVOV + 1                                                 REMOV    6
```

```
      DESTROY REMOV
      LET IS = PSAT(REMOV)
      LET NPOS(IS) = NPOS(IS) - 1
      CALL STATUS(IS,0,9)
      CALL QDMP(IS,0,IL)
      RETURN
      END
      ENDOGENOUS EVENT RETRI
C
CCC   SCHEDULE THE RETRIEVAL OF A SATELLITE BY ENTERING IT INTO
CCC   THE LOADING QUEUE
C
      LET IEVRI = IEVRI + 1
      DESTROY RETRI
      LET RTFLG = 1
      CALL SHIP(PSAT(RETRI),0)
      LET RTFLG = 0
      RETURN
      END
      ENDOGENOUS EVENT SATDN
C
CCC  SATELLITE VOLUNTARILY GOES DOWN AT TERMINATION TIME
C
      LET IEVDN = IEVDN + 1
      LET IS = PSAT(SATDN)
      DESTROY SATDN
      LET MARKS(IS) = 0
      IF SSTAT(IS) EQ OUT, RETURN
      IF NPOS(IS) NE 1, RETURN
      CALL QDMP(IS,0,ILL)
      CALL STATUS(IS,0,3)
      LET MARKS(IS) = 0
      RETURN
      END
      SUBROUTINE SAVER(T2,IS)
      LET IPOL = POLDN(ITSAT(IS))
      LET JSY = ITSYS(IS)
      IF IPOL LT 2, RETURN
      IF IPOL GT 4, RETURN
      IF IPOL EQ 2, GO TO 10
C
CCC   SCHEDULE SATELLITE RETRIEVAL (RETRI) AT TERMINATION TIME +-
C
      IF MARKD(IS) EQ 0, GO TO 1
      CANCEL RETRI CALLED MARKD(IS)
      DESTROY RETRI CALLED MARKD(IS)
      LET MARKD(IS) = 0
    1 LET T = T2 + WAIT2
      IF T LT TIME, GO TO 10
      IF T GT TGOSY(JSY), GO TO 10
      IF T GT TIMES- WSATN, GO TO 10
      CREATE RETRI
      LET PSAT(RETRI) = IS
      CAUSE RETRI AT T
   10 IF IPOL GT 3, GO TO 20
C
C     SCHEDULE NEW SATELLITE (NWSAT) AT TERMINATION TIME +-
```

```
REMOV     7
REMOV     8
REMOV     9
REMOV    10
REMOV    11
REMOV    12
REMOV    13
RETRI     2
RETRI     3
RETRI     4
RETRI     5
RETRI     6
RETRI     7
RETRI     8
RETRI     9
RETRI    10
RETRI    11
RETRI    12
RETRI    13
SATDN     2
SATDN     3
SATDN     4
SATDN     5
SATDN     6
SATDN     7
SATDN     8
SATDN     9
SATDN    10
SATDN    11
SATDN    12
SATDN    13
SATDN    14
SATDN    15
SATDN    16
SAVER     2
SAVER     3
SAVER     4
SAVER     5
SAVER     6
SAVER     7
SAVER     8
SAVER     9
SAVER    10
SAVER    11
SAVER    12
SAVER    13
SAVER    14
SAVER    15
SAVER    16
SAVER    17
SAVER    18
SAVER    19
SAVER    20
SAVER    21
SAVER    22
SAVER    23
SAVER    24
```

```
      IF MARKU(IS) EQ 0, GO TO 2                                        SAVER     25
      CANCEL NWSAT CALLED MARKU(IS)                                     SAVER     26
      DESTROY NWSAT CALLED MARKU(IS)                                    SAVER     27
      LET MARKU(IS) = 0                                                 SAVER     28
    2 LET T = T2                                                        SAVER     29
      IF T GT TGOSY(JSY), GO TO 20                                      SAVER     30
      IF T LT TIME, GO TO 20                                            SAVER     31
      IF T GT TIMES- NSATN, GO TO 20                                    SAVER     32
      CREATE NWSAT                                                      SAVER     33
      LET PSAT(NWSAT) = IS                                              SAVER     34
      CAUSE NWSAT AT T                                                  SAVER     35
   20 RETURN                                                            SAVER     36
      END                                                               SAVER     37
      SUBROUTINE SHIP (IS,IM)                                           SAVER     38
                                                                        SHIP       2
C                                                                       SHIP       3
CCC   THIS IS THE LOADING ROUTINE                                       SHIP       4
C                                                                       SHIP       5
      IF IS GT 0, LET IORB = ORBIT(ITSAT(IS))                           SHIP       6
      LET IFLAG = 0                                                     SHIP       7
      IF RQSEP(IORB) NE 0, LET IFLAG = 1                                SHIP       8
      LET EXVEH = EXORB(IORB)                                           SHIP       9
      IF IS LE 0, GO TO 30                                              SHIP      10
C                                                                       SHIP      11
CCC   ENTER PAYLOAD INTO LOADING QUEUE AND TRY FLIGHT                   SHIP      12
C                                                                       SHIP      13
    1 CALL PAYLQ(IS,IM,ILL)                                             SHIP      14
      IF ILL NE 0, RETURN                                               SHIP      15
    5 CALL GETV(IGO)                                                    SHIP      16
      IF IGO NE 0, GO TO 50                                             SHIP      17
      IF NL(IORB) EQ IL, GO TO 15                                       SHIP      18
      CALL MARKQ                                                        SHIP      19
      IF NQ EQ 0, RETURN                                                SHIP      20
      CALL PROP(MARKP)                                                  SHIP      21
      IF NQ LT 0, GO TO 10                                              SHIP      22
      LET EXORB(IORB) = EXVEH                                           SHIP      23
      IF W(IORB) GE 0., RETURN                                          SHIP      24
C                                                                       SHIP      25
CCC    LAUNCH PAYLOADS FROM QUEUE - SLOAD                               SHIP      26
C                                                                       SHIP      27
   10 IF NL(IORB) EQ 0, GO TO 30                                        SHIP      28
      IF IGO NE 0, GO TO 80                                             SHIP      29
      IF ISEPS EQ 0, GO TO 12                                           SHIP      30
      IF NQ LT 0, GO TO 12                                              SHIP      31
      LET NQ = NL(IORB)                                                 SHIP      32
      LET ILOAD(1) = PQUE(IORB)                                         SHIP      33
      LET ILOAD(J+1) = CITEM(ILOAD(J)),FOR J=(1)(NQ-1)                  SHIP      34
      IF NQ LT 0, GO TO 12                                              SHIP      35
      CALL PROP(MARKP)                                                  SHIP      36
      IF W(IORB) LT 0., GO TO 11                                        SHIP      37
      IF MARKP NE 0, GO TO 12                                           SHIP      38
      IF NEXIT(ISEPS) EQ 2, GO TO 11                                    SHIP      39
      IF NEXIT(ISEPS) NE 6, GO TO 12                                    SHIP      40
   11 LET DUMMY = 0                                                     SHIP      41
      LET ISEPS = 0                                                     SHIP      42
      LET IFLAG = 0                                                     SHIP      43
      GO TO 21                                                          SHIP      44
```

```
12    LET DUMMY = 0
      CALL ISPAY(WGH,WGHDN)
      CALL ISVEH(WGH,WGHDN)
      LET W(IORB) = 0
      LET NL(IORB) = 0
      LET EXVEH = EXORB(IORB)
      IF RQSEP(IORB) EQ 0, RETURN
      LET EXORB(IORB) = 0
      LET EXVEH = 0
      LET IFLAG = 1
      GO TO 5
C
CC    HEAVY PAYLOAD IN LOADING QUEUE
C
   31 LET J = FORBQ(IORB)
      IF W(IORB) EQ 0., RETURN
      LET NX = ISAT(J)
      LET NM = IMOD(J)
      IF EXVEH EQ 0, GO TO 60
C
CC    PAYLOAD EXTREMELY HEAVY - LIMIT TO 1 MONTE CARLO CYCLE
C
   32 CALL STATUS(NX,NM,7)
      CALL DROPQ(J,IORB)
      LET EXVEH = 0
      LET EXORB(IORB) = 0
      IF QUIT EQ 0, LET TRIGS = 1
C
CCC   LOADING ALGORITHM
C
   30 IF ORBQ(IORB) IS EMPTY, RETURN
      CALL GETV(IGO)
      IF IGO EQ 0, GO TO 21
      IF RQSEP(IORB) EQ 0, GO TO 21
      IF IFLAG EQ 0, GO TO 20
      IF LQTIM(FORBQ(IORB)) GT 1000., RETURN
      IF IGO NE 3, RETURN
      LET IGO = 0
      LET ISEPS = 0
20    LET DUMMY = 0
21    LET I = 1
      LET NL(IORB) = 0
   34 LET W(IORB) = 0
      DO TO 40, FOR ALL PAYLD IN ORBQ(IORB)
      IF LQTIM(PAYLD) LT 3000., GO TO 35
      IF LQTIM(FORBQ(IORB)) GT 1000., RETURN
   35 LET NQ = I
      LET ILOAD(NQ) = PAYLD
      CALL PROP(MARKP)
      IF NQ LT 3,     GO TO 10
      LET EXORB(IORB) = EXVEH
      IF W(IORB) LT 0., GO TO 50
      LET I = I + 1
      IF I GT IL, GO TO 10
   40 LOOP
      IF LQTIM(FORBQ(IORB)) LT 1000., GO TO 10
      IF IS GT 0, RETURN
```

```
SHIP      45
SHIP      46
SHIP      47
SHIP      48
SHIP      49
SHIP      50
SHIP      51
SHIP      52
SHIP      53
SHIP      54
SHIP      55
SHIP      56
SHIP      57
SHIP      58
SHIP      59
SHIP      60
SHIP      61
SHIP      62
SHIP      63
SHIP      64
SHIP      65
SHIP      66
SHIP      67
SHIP      68
SHIP      69
SHIP      70
SHIP      71
SHIP      72
SHIP      73
SHIP      74
SHIP      75
SHIP      76
SHIP      77
SHIP      78
SHIP      79
SHIP      80
SHIP      81
SHIP      82
SHIP      83
SHIP      84
SHIP      85
SHIP      86
SHIP      87
SHIP      88
SHIP      89
SHIP      90
SHIP      91
SHIP      92
SHIP      93
SHIP      94
SHIP      95
SHIP      96
SHIP      97
SHIP      98
SHIP      99
SHIP     100
SHIP     101
```

```
41 LET NL(IORB) = 0                                                    SHIP      102
   LET WT(IORB) = 0                                                    SHIP      103
   RETURN                                                              SHIP      104
50 IF NL(IORB) NE 0, GO TO 10                                          SHIP      105
   GO TO 31                                                            SHIP      106
C                                                                      SHIP      107
CC    UPGRADE VEHICLE TO EXPEND MODE                                   SHIP      108
CC                                                                     SHIP      109
60    IF IFLAG NE 0, GO TO 70                                          SHIP      110
      LET EXVEH = 1                                                    SHIP      111
      LET EXORB(IORB) = 1                                              SHIP      112
      LET NL(IORB) = 0                                                 SHIP      113
      CALL STATUS(NX,NM,10)                                            SHIP      114
      GO TO 21                                                         SHIP      115
70    IF NOTUG NE 0, GO TO 32                                          SHIP      116
      LET ISEPS = 0                                                    SHIP      117
      IF IGO EQ 3, LET IGO = 0                                         SHIP      118
      LET NL(IORB) = 0                                                 SHIP      119
      LET IFLAG = 0                                                    SHIP      120
      GO TO 21                                                         SHIP      121
80    IF IGO NE 3, RETURN                                              SHIP      122
      LET NQ = NL(IORB)                                                SHIP      123
      IF NQ LT 2, GO TO 85                                             SHIP      124
      LET ILOAD(1) = PQUE(IORB)                                        SHIP      125
      LET ILOAD(J+1) = CITEM(ILOAD(J)), FOR J = (1)(NQ-1)              SHIP      126
      CALL PROP(MARKP)                                                 SHIP      127
      IF MARKP NE 0, GO TO 85                                          SHIP      128
      IF NEXIT(ISEPS) NE 6, GO TO 85                                   SHIP      129
      LET ISEPS = 0                                                    SHIP      130
      LET IGO = 0                                                      SHIP      131
      LET IFLAG = 0                                                    SHIP      132
      GO TO 21                                                         SHIP      133
85    LET DUMMY = 0                                                    SHIP      134
      IF NOTUG NE 0, RETURN                                            SHIP      135
      IF IS GT 0, RETURN                                               SHIP      136
      IF LQTIM(FORBQ(IORB)) GT 1000., RETURN                           SHIP      137
      LET IGO = 0                                                      SHIP      138
      LET ISEPS = 0                                                    SHIP      139
      LET IFLAG = 0                                                    SHIP      140
      GO TO 21                                                         SHIP      141
      END                                                              SHIP      142
      ENDOGENOUS EVENT START                                           START       2
C                                                                      START       3
CCC   THIS ROUTINE WILL INITIALIZE EACH MONTE CARLO CYCLE              START       4
C                                                                      START       5
      DESTROY START                                                    START       6
      IF TRIG EQ 0, WRITE ON 6                                         START       7
      FORMAT(*1*,S27,*CHRONOLOGICAL TIME HISTORY OF BASE CYCLE*/S5,*TIME START       8
     *            SYSTEM     STATUS    SATELLITE STATUS    MODULE    STATUS START       9
     *    VEHICLE    STATUS*)                                          START      10
      IF TRIG EQ 0, WRITE ON 6, TIME                                   START      11
      FORMAT(* *,S4,M5.2.2,S3,*START SIMULATION*,//)                   START      12
C                                                                      START      13
CCC   SET UP EVENTS FOR NEW SATELLITE LAUNCHES                         START      14
C                                                                      START      15
      DO TO 10, FOR ALL NEW IN NEWS                                    START      16
      LET IS = SCHSY(NEW)                                              START      17
```

```
      LET DTIME(IS) = 0                                                          START      18
      LET STAT(ITSYS(IS)) = DOWN                                                 START      19
      LET SSTAT(L) = DOWN, FOR L=(FSAT(ITSYS(IS)))(LSAT(ITSYS(IS)))              START      20
      IF SCHDT(NEW) LT TIMEB, GO TO 10                                           START      21
      CREATE NWSAT                                                               START      22
      LET PSAT(NWSAT) = SCHSY(NEW)                                               START      23
      LET PMOD(NWSAT) = 0                                                        START      24
      CAUSE NWSAT AT SCHDT(NEW)                                                  START      25
   10 LOOP                                                                       START      26
      LET MSEP(I) = 0, FOR I=(1)(NSEPS)                                          START      27
      LET NEXIT(I) = 0, FOR I=(1)(NSEPS)                                         START      28
      DO TO 2, FOR I=(1)(SYORB)                                                  START      29
      LET SATLF(I) = 0                                                           START      30
      LET LFSAT(I) = 0                                                           START      31
      LET BEGST(I) = 0                                                           START      32
      LET TLAST(I) = 0                                                           START      33
      LET SDTST(I) = 0                                                           START      34
      LET NPOS(I) = 0                                                            START      35
      IF MOD(I) IS EMPTY, GO TO 2                                                START      36
      DO TO 1, FOR ALL MODSY IN MOD(I)                                           START      37
      LET NUM  (MODSY) = 0                                                       START      38
      LET LOADF(MODSY) = 0                                                       START      39
      LET MSTAT(MODSY) = 0                                                       START      40
    1 LOOP                                                                       START      41
    2 LOOP                                                                       START      42
      LET VSHUT(I) = 1, FOR I=(1)(NSHUT)                                         START      43
      LET VTUG(I) = 1, FOR I=(1)(NTUG)                                           START      44
      LET VPAD(I) = 1, FOR I=(1)(NPAD)                                           START      45
      LET SUTFY(I) = 0, FOR I=(1)(NYEAR)                                         START      46
      LET SEPFY(I) = 0, FOR I=(1)(NYEAR)                                         START      47
      LET VSEPS(I) = 1, FOR I=(1)(NSEPS)                                         START      48
      LET AVSEP(I) = 0., FOR I=(1)(NSEPS)                                        START      49
      LET SWDN(I) = 0, FOR I=(1)(NSEPS)                                          START      50
      LET SLDN(I) = 0, FOR I=(1)(NSEPS)                                          START      51
      LET TUGFY(I) = 0, FOR I=(1)(NYEAR)                                         START      52
      LET CVA(I) = 0., FOR I=(1)(3)                                              START      53
      LET TGO(I) = 0., FOR I=(1)(SYORB)                                          START      54
      LET TGOSY(I) = 0., FOR I=(1)(STSTB)                                        START      55
      LET BEGSY(I) = 0., FOR I=(1)(STSTB)                                        START      56
      LET TLASY(I) = 0., FOR I=(1)(STSTB)                                        START      57
      LET SDTSY(I) = 0., FOR I=(1)(STSTB)                                        START      58
      IF MODB NE MODS, LET EXMOD = MODB                                          START      59
      IF MODB EQ MODS, LET MODS = EXMOD                                          START      60
      LET MDCNT(I) = 0, FOR I=(1)(MITAB)                                         START      61
      LET NOWAR(I) = 0, FOR I=(1)(MITAB)                                         START      62
      LET NOFAL(I) = 0, FOR I=(1)(MITAB)                                         START      63
      IF LIMIT EQ 0., LET LIMIT = 20000.                                         START      64
      LET EXORB(I) = 0, FOR I=(1)(NORBS)                                         START      65
C                              ******************************************        START      66
C     REINITIALIZE NOMOD ON ALL SATELLITES                                       START      67
C     CREATE NEWME EVENTS                                                        START      68
C                              ******************************************        START      69
C                                                                                START      70
C        SET UP END OF MONTE CARLO CYCLE EVENT                                   START      71
C                                                                                START      72
         CREATE TERM                                                             START      73
         CAUSE TERM AT 3000.                                                     START      74
```

```
      RETURN                                                                    START      75
      END                                                                       START      76
      SUBROUTINE STATUS(IS,IM,IST)                                              STATUS      2
C                                                                               STATUS      3
CCC   IM = 0, SATELLITE                                                         STATUS      4
CCC   IM = +, REPLACEABLE MODULE                                                STATUS      5
C                                                                               STATUS      6
CCC   IST = 1, AVAILABLE                                                        STATUS      7
CCC   IST = 2, UP                                                               STATUS      8
CCC   IST = 3, DOWN                                                             STATUS      9
CCC   IST = 4, LAUNCHED                                                         STATUS     10
CCC   IST = 5, ME UPGRADE                                                       STATUS     11
CCC   IST = 6, SATELLITE RETRIEVED                                              STATUS     12
CCC   IST = 7, PAYLOAD IS TOO HEAVY, NOT FLOWN - DROPPED FROM QUEUE             STATUS     13
CCC   IST = 8, WARNING ON MODULE                                                STATUS     14
CCC   IST = 9, SATELLITE REMOVED FROM ORBIT                                     STATUS     15
C                                                                               STATUS     16
         LET JST = ITSAT(IS)                                                    STATUS     17
         LET JSY = ITSYS(IS)                                                    STATUS     18
         IF IM NE 0, LET JMD = NOMOD(IM)                                        STATUS     19
         LET HALST(IS) = TIME                                                   STATUS     20
         LET HALSY(JSY) = TIME                                                  STATUS     21
         IF IST EQ 2, LET ISTAT = UP                                            STATUS     22
         IF IST EQ 3, LET ISTAT = DOWN                                          STATUS     23
         IF TRIG2 EQ 1, GO TO 54                                                STATUS     24
         LET DELTA = 0                                                          STATUS     25
         GO TO (10,8,8,10,10,10,10,2,4,10),IST                                  STATUS     26
       8 IF IM EQ 0, GO TO 5                                                    STATUS     27
         LET MSTAT(IM) = IST                                                    STATUS     28
         IF IST EQ 2, GO TO 2                                                   STATUS     29
C                                                                               STATUS     30
CCC      NRU FAILURE-SCHEDULE NWSAT                                             STATUS     31
C                                                                               STATUS     32
         CALL REDUN(IS,IM)                                                      STATUS     33
         LET FREE = DELTA/3000.                                                 STATUS     34
         LET IK = NRU(IM)                                                       STATUS     35
         IF DELTA NE 0., GO TO 111                                              STATUS     36
         IF EXMOD NE 0, LET IK = EXMOD                                          STATUS     37
         IF XSAT(IS) NE 0, LET IK=XSAT(IS)                                      STATUS     38
     111 IF IK NE 100, GO TO 1                                                  STATUS     39
         DO TO 200, FOR ALL MODSY IN MOD(IS)                                    STATUS     40
         CALL QDMP(IS,MODSY,ILL)                                                STATUS     41
     200 LOOP                                                                   STATUS     42
         LET SSTAT(IS) = OUT                                                    STATUS     43
C                                                                               STATUS     44
CCC      TEST LAUNCH POLICY ON NRU FAILURE                                      STATUS     45
C                                                                               STATUS     46
         IF PDOWN EQ 0, GO TO 1                                                 STATUS     47
         LET T = TIME + WAIT3                                                   STATUS     48
         CALL SAVER(T,IS)                                                       STATUS     49
       1 IF DELTA NE 0., GO TO 7                                                STATUS     50
         IF SSTAT(IS) NE OUT, LET SSTAT(IS) = ISTAT                             STATUS     51
         GO TO 7                                                                STATUS     52
       2 IF SSTAT(IS) EQ OUT, GO TO 10                                          STATUS     53
         GO TO 6                                                                STATUS     54
       4 LET ISTAT = SSTAT(IS)                                                  STATUS     55
         IF NPOS(IS) EQ 0, LET ISTAT = OUT                                      STATUS     56
```

```
      LET SSTAT(IS) = ISTAT                                                  STATUS    57
      GO TO 7                                                                STATUS    58
    5 LET SSTAT(IS) = ISTAT                                                  STATUS    59
      LET MSTAT(MODSY) = IST, FOR ALL MODSY IN MOD(IS)                       STATUS    60
      IF SSTAT(IS) EQ DOWN, LET SSTAT(IS) = OUT                              STATUS    61
      IF NPOS(IS) EQ 0, LET SSTAT(IS) = OUT                                  STATUS    62
      IF SSTAT(IS) EQ OUT, GO TO 7                                           STATUS    63
    6 DO TO 38, FOR ALL MODSY IN MOD(IS)                                     STATUS    64
      IF MSTAT(MODSY) EQ 2, GO TO 38                                         STATUS    65
      CALL REDUN(IS,MODSY)                                                   STATUS    66
      IF DELTA EQ 0., GO TO 7                                                STATUS    67
   38 LOOP                                                                   STATUS    68
      LET SSTAT(IS) = UP                                                     STATUS    69
    7 LET K = 0                                                              STATUS    70
      LET KK = 0                                                             STATUS    71
      IF TRIG + TRIG2 EQ 1, GO TO 10                                         STATUS    72
      DO TO 39, FOR I=(FSAT(JSY))(LSAT(JSY))                                 STATUS    73
      IF SSTAT(I) EQ UP, LET K = K + 1                                       STATUS    74
      IF SSTAT(I) EQ DOWN, LET KK = KK +1                                    STATUS    75
   39 LOOP                                                                   STATUS    76
      LET IPOL = POLON(JST)                                                  STATUS    77
      LET IT = DOWN                                                          STATUS    78
      IF IPOL EQ 0, LET IT = OUT                                             STATUS    79
      IF IPOL EQ 1, LET IT = OUT                                             STATUS    80
      IF IPOL EQ 4, LET IT = OUT                                             STATUS    81
      IF TIME GE TGOSY(JSY), LET IT = OUT                                    STATUS    82
      IF K NE 0, LET IT = DOWN                                               STATUS    83
      IF KK NE 0, LET IT = DOWN                                              STATUS    84
      LET STAT(JSY)= IT                                                      STATUS    85
      IF K GE NFUP(JSY), LET STAT(JSY) = UP                                  STATUS    86
   10 CALL OUTAG(IS,JSY)                                                     STATUS    87
   54 IF TRIG NE 0, RETURN                                                   STATUS    88
      IF TIME LT TIMEB, RETURN                                               STATUS    89
      CALL STPRT(IS,IM,JSY,JST,JMD,ISTAT,IST)                                STATUS    90
      RETURN                                                                 STATUS    91
      END                                                                    STATUS    92
      SUBROUTINE OUTAG(IS,JSY)                                               OUTAG      2
C                                                                            OUTAG      3
CC    DO COMPUTATIONS FOR SATELLITE AND SYSTEM                               OUTAG      4
CC    AVAILABILITIES AND DELAY TO RESTORE INTERVALS                          OUTAG      5
C                                                                            OUTAG      6
      IF TLAST(IS) EQ 0., GO TO 54                                           OUTAG      7
      IF SSTAT(IS) EQ UP, GO TO 51                                           OUTAG      8
      IF TLAST(IS) LT 0., GO TO 52                                           OUTAG      9
      LET SDTST(IS) = SDTST(IS) + TIME - TLAST(IS)                           OUTAG     10
      LET TLAST(IS) = -TIME                                                  OUTAG     11
      GO TO 52                                                               OUTAG     12
   51 IF TLAST(IS) GT 0., GO TO 52                                           OUTAG     13
      LET A = TIME + TLAST(IS)                                               OUTAG     14
      LET TLAST(IS) = TIME                                                   OUTAG     15
      IF A EQ 0., GO TO 52                                                   OUTAG     16
      LET DNTST(IS) = DNTST(IS) + A                                          OUTAG     17
      LET C223(IS) = C223(IS) + 1.                                           OUTAG     18
      IF N223(IS) GT A, LET N223(IS) = A                                     OUTAG     19
      IF X223(IS) LT A, LET X223(IS) = A                                     OUTAG     20
C                                                                            OUTAG     21
CC    SYSTEM STATUS                                                          OUTAG     22
```

```
C
   52 LET IY = JSY
      IF TGOSY(IY) EQ 0, GO TO 54
      IF STAT(IY) EQ UP, GO TO 53
      IF TLASY(IY) LT 0., GO TO 54
      LET SDTSY(IY) = SDTSY(IY) + TIME - TLASY(IY)
      LET TLASY(IY) = -TIME
      GO TO 54
   53 IF TLASY(IY) GT 0., GO TO 54
      LET A = TIME + TLASY(IY)
      LET TLASY(IY) = TIME
      IF A EQ 0., GO TO 54
      LET DNTSY(IY) = DNTSY(IY) + A
      LET C208(IY) = C208(IY) + 1.
      IF  N208(IY) GT A, LET N208(IY) = A
      IF  X208(IY) LT A, LET X208(IY) = A
   54 RETURN
      END
      SUBROUTINE STPRT(IS,IM,JSY,JST,JMD,ISTAT,IST)
C
CCC   PRINT STATUS LINE FOR ALL OPTIONS
C
      IF TRIG2 EQ 0, CALL FILES(IS,IM,IST)
      IF TRIG2 EQ 0, WRITE ON 6
      FORMAT (S1)
      LET IP = IS - FSAT(JSY) + 1
      LET NSY = SYNAM(JSY)
      LET NSS = STAT(JSY)
      LET KST = SNAME(JST)
      LET KSS = SSTAT(IS)
      LET TE = TIME
      LET I = DPART(TE)
      LET J = HPART(TE) + 1
      LET K = MPART(TE) + 1
      IF IM EQ 0, GO TO (11,12,12,14,14,16,17,17,18,19), IST
      LET MST = MNAME(JMD)
      LET MN = MNO(IM)
      GO TO (21,22,22,24,25,25,27,26,22,29),IST
C
CCC   PRINT SATELLITE STATUS
C
   11 WRITE ON 6,I,J,K,NSY,NSS,IP,KST
      FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,*AVAILABLE*)
      RETURN
   12 WRITE ON 6,I,J,K,NSY,NSS,IP,KST,KSS
      FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,A6)
      RETURN
   14 WRITE ON 6,I,J,K,NSY,NSS,IP,KST
      FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,*LAUNCHED*)
      RETURN
   16 WRITE ON 6,I,J,K,NSY,NSS,IP,KST
      FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,*RETRIEVED*)
      RETURN
   17 WRITE ON 6,I,J,K,NSY,BLANK,IP,KST
      FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,*SATELLITE TOO
     *HEAVY  ++++++++++++++++++++++++++++++++++++++++*)
      RETURN
```

```
OUTAG   23
OUTAG   24
OUTAG   25
OUTAG   26
OUTAG   27
OUTAG   28
OUTAG   29
OUTAG   30
OUTAG   31
OUTAG   32
OUTAG   33
OUTAG   34
OUTAG   35
OUTAG   36
OUTAG   37
OUTAG   38
OUTAG   39
OUTAG   40
STPRT    2
STPRT    3
STPRT    4
STPRT    5
STPRT    6
STPRT    7
STPRT    8
STPRT    9
STPRT   10
STPRT   11
STPRT   12
STPRT   13
STPRT   14
STPRT   15
STPRT   16
STPRT   17
STPRT   18
STPRT   19
STPRT   20
STPRT   21
STPRT   22
STPRT   23
STPRT   24
STPRT   25
STPRT   26
STPRT   27
STPRT   28
STPRT   29
STPRT   30
STPRT   31
STPRT   32
STPRT   33
STPRT   34
STPRT   35
STPRT   36
STPRT   37
STPRT   38
STPRT   39
STPRT   40
```

```
18 WRITE ON 6,1,J,K,NSY,NSS,IP,KST                                      STPRT      41
   FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,*REMOVED*)        STPRT      42
   RETURN                                                               STPRT      43
19 WRITE ON 6,I,J,K,NSY,NSS,IP,KST                                      STPRT      44
   FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,*SATELLITE REQU   STPRT      45
  *IRES EXPENDED VEHICLE----------------------------------*)            STPRT      46
   RETURN                                                               STPRT      47
C                                                                       STPRT      48
CCC   PRINT MODULE STATUS                                               STPRT      49
C                                                                       STPRT      50
21 WRITE ON 6,I,J,K,NSY,NSS,IP,KST,KSS,MN,MST                           STPRT      51
   FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,A6,I3,S1,A6,S4,   STPRT      52
  **AVAILABLE*)                                                         STPRT      53
   RETURN                                                               STPRT      54
22 WRITE ON 6,I,J,K,NSY,NSS,IP,KST,KSS,MN,MST,ISTAT                     STPRT      55
   FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,A6,I3,S1,A6,S4,   STPRT      56
  *A6)                                                                  STPRT      57
   IF FREE EQ 0, RETURN                                                 STPRT      58
   WRITE ON 6                                                           STPRT      59
   FORMAT(*+*,S79,*(FREEBIE)*)                                          STPRT      60
   LET FREE = 0                                                         STPRT      61
   RETURN                                                               STPRT      62
24 WRITE ON 6,I,J,K,NSY,NSS,IP,KST,KSS,MN,MST                           STPRT      63
   FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,A6,I3,S1,A6,S4,   STPRT      64
  **LAUNCHED*)                                                          STPRT      65
   IF FREE EQ 0, RETURN                                                 STPRT      66
   WRITE ON 6                                                           STPRT      67
   FORMAT(*+*,S78,*(FREEBIE)*)                                          STPRT      68
   LET FREE = 0                                                         STPRT      69
   RETURN                                                               STPRT      70
25 WRITE ON 6,I,J,K,NSY,NSS,IP,KST,KSS,MN,MST                           STPRT      71
   FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,A6,I3,S1,A6,S4,   STPRT      72
  **ME UPGRADE*)                                                        STPRT      73
   RETURN                                                               STPRT      74
26 WRITE ON 6,I,J,K,NSY,NSS,IP,KST,KSS,MN,MST                           STPRT      75
   FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,A6,I3,S1,A6,S4,   STPRT      76
  **WARNING*)                                                           STPRT      77
   RETURN                                                               STPRT      78
27 WRITE ON 6,I,J,K,NSY,BLANK,IP,KST,BLANK,MN,MST                       STPRT      79
   FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,A6,I3,S1,A6,S4,   STPRT      80
  **MODULE TOO HEAVY ++++++++++++++++++++++++++++++++++++++++++*)       STPRT      81
   RETURN                                                               STPRT      82
29 WRITE ON 6,I,J,K,NSY,NSS,IP,KST,KSS,MN,MST                           STPRT      83
   FORMAT(S5,I5,*.*,I2,*.*,I2,S3,A6,S4,A6,I3,S1,A6,S4,A6,I3,S1,A6,S4,   STPRT      84
  **MODULE REQUIRES EXPENDED VEHICLE----------------------*)            STPRT      85
   RETURN                                                               STPRT      86
   END                                                                  STPRT      87
   ENDOGENOUS EVENT TERM                                                TERM        2
C                                                                       TERM        3
CCCCCCC   THIS ROUTINE WILL BE ACTIVATED AT THE END OF A MONTE CARLO CYCLE  TERM    4
C                                                                       TERM        5
CCCCCCC   IT MAY RESTART THE PROGRAM FOR THE NEXT CYCLE OR CAUSE THE    TERM        6
C                                                                       TERM        7
CCCCCCC   TERMINATION OF THE RUN WITH STATISTICS.                       TERM        8
C                                                                       TERM        9
   DESTROY TERM                                                         TERM       10
   IF TRIG EQ 0, WRITE ON 6,TIME                                        TERM       11
```

```
      FORMAT(/,S5,M5.2,2,S3,*TERMINATE SIMULATION*)                     TERM       12
      IF TRIG2 EQ 0, CALL FILEO                                         TERM       13
                                                                        TERM       14
C                                                                       TERM       15
CC    CLEAN UP QUEUES AT END OF MONTE CARLO CYCLE                       TERM       16
C                                                                       TERM       17
      LET TRIG = TRIG + 1                                               TERM       18
      DO TO 6, FOR J=(1)(NORBS)                                         TERM       19
      IF ORBQ(J) IS EMPTY, GO TO 6                                      TERM       20
C                                                                       TERM       21
CC    DROP FREEBIES                                                     TERM       22
C                                                                       TERM       23
      LET I = J                                                         TERM       24
      DO TO 20, FOR ALL PAYLD IN ORBQ(I)                                TERM       25
      IF LQTIM(PAYLD) GT 3000., CALL DROPQ(PAYLD,I)                     TERM       26
   20 LOOP                                                              TERM       27
      LET IORB = J                                                      TERM       28
      IF ORBQ(J) IS EMPTY, GO TO 6                                      TERM       29
C                                                                       TERM       30
CC   LOADING QUEUE CONTAINS TRASH -- STOP RUN                           TERM       31
C                                                                       TERM       32
      WRITE ON 6                                                        TERM       33
      FORMAT(S5,*---RUN STOPPED DUE TO DATA IN LOADING QUEUE AT END OF C TERM       34
     *YCLE*)                                                            TERM       35
      DO TO 2, FOR ALL PAYLD IN ORBQ(IORB)                              TERM       36
      LET I = SNAME(ITSAT(ISAT(PAYLD)))                                 TERM       37
      LET A = LQTIM(PAYLD)                                              TERM       38
      IF IMOD(PAYLD) EQ 0, WRITE ON 6,I,A                               TERM       39
      FORMAT(S5,*SATELLITE - *,A6,* SINCE *,M5.2,2)                     TERM       40
      IF IMOD(PAYLD) NE 0, WRITE ON 6,MNAME(NOMOD(IMOD(PAYLD))),I,A     TERM       41
      FORMAT(S5,*MODULE - *,A6,* ON SATELLITE - *,A6,* SINCE *,M5.2.2)  TERM       42
    2 LOOP                                                              TERM       43
      LET TRIGS = 1                                                     TERM       44
    6 LOOP                                                              TERM       45
C                                                                       TERM       46
CC  GATHER MONTE CARLO END OF CYCLE STATISTICS FOR VEHICLES/SATELLITES  TERM       47
C                                                                       TERM       48
   10 CALL MCVEH                                                        TERM       49
      CALL MCMOD                                                        TERM       50
      CALL MCSAT                                                        TERM       51
      CALL MCSYS                                                        TERM       52
      IF TRIG GE TRIGS, GO TO 5                                         TERM       53
C                                                                       TERM       54
CC   INITIALIZE  ANOTHER CYCLE                                          TERM       55
C                                                                       TERM       56
      CREATE START                                                      TERM       57
      LET TIME = 0.                                                     TERM       58
      CAUSE START AT 1.                                                 TERM       59
      IF TRIG GT 1, RETURN                                              TERM       60
      CALL TEREV                                                        TERM       61
      CALL TERV1                                                        TERM       62
      CALL TERV2                                                        TERM       63
      RETURN                                                            TERM       64
C                                                                       TERM       65
CC  FINAL OUTPUT                                                        TERM       66
C                                                                       TERM       67
    5 CALL TERSY                                                        TERM       68
      CALL TEREV
```

```
      CALL TERM2                                                        TERM       69
      CALL TERV2                                                        TERM       70
      CALL TERV1                                                        TERM       71
      STOP                                                              TERM       72
      END                                                               TERM       73
      SUBROUTINE TEREV                                                  TEREV       2
C                                                                       TEREV       3
CCC   PRINTS STATISTICS OF EVENTS                                       TEREV       4
C                                                                       TEREV       5
      LET A = TRIG                                                      TEREV       6
      RETURN                                                            TEREV       7
      END                                                               TEREV       8
      SUBROUTINE TERV1                                                  TERV1       2
C                                                                       TERV1       3
CCCCC                                                                   TERV1       4
   OUTPUT STATISTICS FOR FLIGHTS PER YEAR                               TERV1       5
C                                                                       TERV1       6
C                                                                       TERV1       7
      WRITE ON 6, TRIG,TIMEB,TIMES                                      TERV1       8
      FORMAT(*1*,S7,*STATISTICAL SUMMARY FOR*,I4,* MONTE CARLO CYCLES FO TERV1       9
     *R THE YEARS*,D5,* TO*,D5/)                                        TERV1      10
      LET A = TRIG                                                      TERV1      11
      WRITE ON 6                                                        TERV1      12
      FORMAT(S25,*FLIGHT SUMMARY*/S18,*SHUTTLE*,S15,*TUG*,S17,*SEPS*/   TERV1      13
     **    YEAR      MIN    AVG    MAX      MIN    AVG    MAX      MIN    AVG  TERV1      14
     **MAX*)                                                            TERV1      15
      DO TO 10, FOR I=(1)(NYEAR)                                        TERV1      16
      LET TI = I - 1                                                    TERV1      17
      LET J = TIMEB + TI                                                TERV1      18
      IF MAX90(I) EQ 0, GO TO 10                                        TERV1      19
      LET B = SUM39(I)                                                  TERV1      20
      LET B = B/A                                                       TERV1      21
      LET C = SUM90(I)                                                  TERV1      22
      LET C = C/A                                                       TERV1      23
      LET D = SUM86(I)                                                  TERV1      24
      LET D = D/A                                                       TERV1      25
      WRITE ON 6, J, MIN90(I) ,C,MAX90(I),MIN39(I),B,MAX39(I),          TERV1      26
     * MIN86(I),D,MAX86(I)                                             TERV1      27
      FORMAT(I8,I8,D4.1,I6,I8,D4.1,I6,I8,D4.1,I6)                       TERV1      28
   10 LOOP                                                              TERV1      29
      LET B = ITFLT                                                     TERV1      30
      LET B = B/A                                                       TERV1      31
      LET C = IFSUT                                                     TERV1      32
      LET C = C/A                                                       TERV1      33
      LET D = IFSEP                                                     TERV1      34
      LET D = D/A                                                       TERV1      35
      WRITE ON 6,NFSUT,C,MFSUT,NTFLT,B,MTFLT,NFSEP,D,MFSEP              TERV1      36
      FORMAT(* PROGRAM*,I8,D4.1,I6,I8,D4.1,I6,I8,D4.1,I6)               TERV1      37
      DO TO 5, FOR I=(1)(3)                                             TERV1      38
      IF TRIG LT TRIGS, GO TO 5                                         TERV1      39
C                                                                       TERV1      40
CCC   FIX LATER ***************************************************** TERV1      41
C                                                                       TERV1      42
      IF MTD(I) EQ 1000., LET MTD(I) = 0.                               TERV1      43
      IF MCVA(I) EQ 1000., LET MCVA(I) = 0.                             TERV1      44
      IF TCVA(I) EQ 0., GO TO 5                                         TERV1      45
      LET VTD(I) = VTD(I)*360./TCVA(I)                                  TERV1      46
```

```
      LET TCVA(I) = TCVA(I)/A                                                    TERV1      47
      LET MTD(I) = MTD(I)*360.                                                   TERV1      48
      LET XTD(I) = XTD(I)*360.                                                   TERV1      49
      IF I EQ 1, LET E = C                                                       TERV1      50
      IF I EQ 2, LET E = B                                                       TERV1      51
      IF I EQ 3, LET E = D                                                       TERV1      52
      LET TCVA(I) = TCVA(I)*100./E                                               TERV1      53
      LET MCVA(I) = MCVA(I)*100./E                                               TERV1      54
      LET XCVA(I) = XCVA(I)*100./E                                               TERV1      55
    5 LOOP                                                                       TERV1      56
      WRITE ON 6,MCVA(1),TCVA(1),XCVA(1),MCVA(2),TCVA(2),XCVA(2),                TERV1      57
     * MCVA(3),TCVA(3),XCVA(3)                                                   TERV1      58
      FORMAT(*0PERCENT*,D6.1,2D4.1,D6.1,2D4.1,D6.1,2D4.1)                        TERV1      59
      WRITE ON 6,MTD(1),VTD(1),XTD(1),MTD(2),VTD(2),XTD(2),                      TERV1      60
     * MTD(3),VTD(3),XTD(3)                                                      TERV1      61
      FORMAT(*0 DELAY *,D6.1,2D4.1,D6.1,2D4.1,D6.1,2D4.1)                        TERV1      62
      LET EX = EXTUG/A                                                           TERV1      63
      IF EXTUG NE 0., WRITE ON 6, EX                                             TERV1      64
      FORMAT(*0     AVERAGE NO. OF EXPENDED TUGS = *,D5.1)                       TERV1      65
      RETURN                                                                     TERV1      66
      END                                                                        TERV1      67
      SUBROUTINE TERV2                                                           TERV2       2
C                                                                                TERV2       3
CCC   OUTPUT STATISTICS FOR AVERAGE WEIGHT DELIVERED TO ORBIT                    TERV2       4
C                                                                                TERV2       5
      WRITE ON 6                                                                 TERV2       6
      FORMAT(*1*,S30,*ORBIT TRAFFIC SUMMARY*/*0*,S13,*AVERAGE FLIGHTS*,S         TERV2       7
     *15,*AVERAGE UP WEIGHT*,S9,*SHUTTLE ONLY*/S3,*ORBIT   SHUTTLE    TU        TERV2       8
     *G     SEPS        SHUTTLE        TUG        SEPS     LOAD FACTOR*/S1)     TERV2       9
      LET A = TRIG                                                               TERV2      10
      DO TO 30, FOR I=(1)(NOR3S)                                                 TERV2      11
      IF ORBID(I) EQ 0, GO TO 30                                                 TERV2      12
      LET C = 0.                                                                 TERV2      13
      LET D = 0.                                                                 TERV2      14
      LET E = 0.                                                                 TERV2      15
      LET FB = 0.                                                                TERV2      16
      LET FC = 0.                                                                TERV2      17
      LET FD = 0.                                                                TERV2      18
      IF WSHUT(I) NE 0., LET C = WSHUT(I)/CSHUT(I)                               TERV2      19
      IF WSEPS(I) NE 0., LET D = WSEPS(I)/CSEPS(I)                               TERV2      20
      IF WTUG (I) NE 0., LET E = WTUG (I)/CTUG (I)                               TERV2      21
      IF WDSUT(I) NE 0., LET FB = WDSUT(I)/CDSUT(I)                              TERV2      22
      IF WDSEP(I) NE 0., LET FC = WDSEP(I)/CDSEP(I)                              TERV2      23
      IF WDTUG(I) NE 0., LET FD = WDTUG(I)/CDTUG(I)                              TERV2      24
      LET CSHUT(I) = CSHUT(I)/A                                                  TERV2      25
      LET CSEPS(I) = CSEPS(I)/A                                                  TERV2      26
      LET CTUG (I) = CTUG (I)/A                                                  TERV2      27
      LET CDSUT(I) = CDSUT(I) /A                                                 TERV2      28
      LET CDSEP(I) = CDSEP(I) /A                                                 TERV2      29
      LET CDTUG(I) = CDTUG(I) /A                                                 TERV2      30
      LET J = RQSUT(IORB)                                                        TERV2      31
      IF J EQ 0, LET J = 1                                                       TERV2      32
      LET B = WSHUT(I)/WCONV(J)                                                  TERV2      33
      WRITE ON 6,ORBID(I),CSHUT(I),CDSUT(I),CTUG(I),CDTUG(I),CSEPS(I),           TERV2      34
     * CDSEP(I),C,FB,E,FD,D,FC,B                                                 TERV2      35
      FORMAT(S3,A6,D4.1,*/*,2D4.1,*/*,2D4.1,*/*,D4.1,D12.1,*/*,2D6.1,            TERV2      36
     * */*;2D6.1,*/*,D6.1,D9.2)                                                  TERV2      37
```

```
 30 LOOP                                                                TERV2   38
    RETURN                                                              TERV2   39
    END                                                                 TERV2   40
    SUBROUTINE TERSY                                                    TERSY    2
C                                                                       TERSY    3
CC  OUTPUT STATISTICS FOR SYSTEMS/SATELLITES                            TERSY    4
C                                                                       TERSY    5
    WRITE ON 6                                                          TERSY    6
    FORMAT(*1*)                                                         TERSY    7
    LET A = TRIG                                                        TERSY    8
    LET TSATS = 0.                                                      TERSY    9
    LET EQSAT = 0.                                                      TERSY   10
    DO TO 13, FOR I=(1)(STSTB)                                          TERSY   11
    IF SYNAM(I) EQ 0, GO TO 13                                          TERSY   12
    IF FSAT(I) EQ 0, GO TO 13                                           TERSY   13
    WRITE ON 6,SYNAM(I)                                                 TERSY   14
    FORMAT(*0    STATISTICS FOR SYSTEM - *,A6)                          TERSY   15
    DO TO 12, FOR J=(FSAT(I))(LSAT(I))                                  TERSY   16
    LET TRES = 0.                                                       TERSY   17
    LET ICEQ = 0                                                        TERSY   18
    IF SORTE(ITSAT(J)) NE 0., GO TO 110                                 TERSY   19
    IF MOD(J) IS EMPTY, GO TO 12                                        TERSY   20
    WRITE ON 6                                                          TERSY   21
    FORMAT(*0 MODULE   MIN   AVG   MAX  MIN FLT  AVG FLT  MAX FLT*)     TERSY   22
    DO TO 11, FOR ALL MODSY IN MOD(J)                                   TERSY   23
    LET B = SUMNU(MODSY)                                                TERSY   24
    LET B = B/A                                                         TERSY   25
    IF NRU(MODSY) NE 100, LET ICEQ = ICEQ +1                            TERSY   26
    LET TRES = TRES + B                                                 TERSY   27
    LET D = MINLF(MODSY)                                                TERSY   28
    LET E = SUMLF(MODSY)                                                TERSY   29
    LET E = E/A                                                         TERSY   30
    LET F = MAXLF(MODSY)                                                TERSY   31
    LET D = D/100.                                                      TERSY   32
    LET E = E/100.                                                      TERSY   33
    LET F = F/100.                                                      TERSY   34
    IF MAXNU(MODSY) EQ 0, WRITE ON 6, MNAME(NOMOD(MODSY))               TERSY   35
   *,NRU(MODSY)                                                         TERSY   36
    FORMAT(S3,A6,I3)                                                    TERSY   37
    IF MAXNU(MODSY) NE 0, WRITE ON 6, MNAME(NOMOD(MODSY)),              TERSY   38
   *NRU(MODSY),                                                         TERSY   39
   *  MINNU(MODSY),B,                                                  TERSY   40
   *  MAXNU(MODSY),D,E,F                                               TERSY   41
    FORMAT(S3,A6,2I3,D4.1,I5,3D5.2)                                     TERSY   42
 11 LOOP                                                                TERSY   43
110 LET S227(J) = S227(J)/A                                             TERSY   44
    LET B = NDEP(J)                                                     TERSY   45
    LET B = B/A                                                         TERSY   46
    WRITE ON 6,SNAME(ITSAT(J)),B,N227(J),S227(J),X227(J)                TERSY   47
    FORMAT(*  SATELLITE*/S3,A6,S6,D4.1,S6,3D5.2)                        TERSY   48
    IF SORTE(ITSAT(J)) NE 0., GO TO 12                                  TERSY   49
    LET TSATS = TSATS + B                                               TERSY   50
    LET E = ICEQ                                                        TERSY   51
    IF E NE 0., LET TRES = TRES/E                                       TERSY   52
    LET TRES =TRES+3                                                    TERSY   53
    WRITE ON 6, TRES                                                    TERSY   54
    FORMAT(*   EQ SAT*,S6,D4.2)                                         TERSY   55
```

```
      LET EQSAT = EQSAT + TRES                                          TERSY      56
      LET SUMSL(J) = SUMSL(J)/A                                         TERSY      57
      WRITE ON 6,MINSL(J),SUMSL(J),MAXSL(J)                             TERSY      58
      FORMAT(*0 SATELLITE TOTAL FLIGHTS  *,3D5.2)                       TERSY      59
      IF C223(J) EQ 0., LET C223(J) = 1.                                TERSY      60
      LET F = PERST(J) /A                                               TERSY      61
      LET E = DNTST(J) *360./C223(J)                                    TERSY      62
      IF N223(J) EQ 1000., LET N223(J) = 0.                             TERSY      63
      LET N223(J) = N223(J)*360.                                        TERSY      64
      LET X223(J) = X223(J)*360.                                        TERSY      65
      WRITE ON 6,N216(J),F,X216(J)                                      TERSY      66
      FORMAT(*0 PERCENT SATELLITE AVAIL. *,3D5.2)                       TERSY      67
      WRITE ON 6,N223(J),E,X223(J)                                      TERSY      68
      FORMAT(*0 DELAY INTERVAL TO RESTORE*,3D5.2)                       TERSY      69
   12 LOOP                                                              TERSY      70
      LET SYLF(I) = SYLF(I)/A                                           TERSY      71
      WRITE ON 6,NSYLF(I),SYLF(I),XSYLF(I)                              TERSY      72
      FORMAT(*0 SYSTEM TOTAL FLIGHTS     *,3D5.2)                       TERSY      73
      IF FSAT(I) EQ 0, GO TO 13                                         TERSY      74
      LET F = PERSY(I)/A                                                TERSY      75
      IF C208(I) EQ 0., LET C208(I) = 1.                                TERSY      76
      LET E = DNTSY(I)*360./C208(I)                                     TERSY      77
      IF N208(I) EQ 1000., LET N208(I) = 0.                             TERSY      78
      LET N208(I) = N208(I)*360.                                        TERSY      79
      LET X208(I) = X208(I)*360.                                        TERSY      80
      IF X200(I) EQ 0., GO TO 16                                        TERSY      81
      WRITE ON 6,N200(I),F,X200(I)                                      TERSY      82
      FORMAT(*0 PERCENT SYSTEM AVAILABLE *,3D5.2)                       TERSY      83
      WRITE ON 6,N208(I),E,X208(I)                                      TERSY      84
      FORMAT(*0 DELAY INTERVAL TO RESTORE*,3D5.2)                       TERSY      85
   16 WRITE ON 6                                                        TERSY      86
      FORMAT(*----------------------------------------------------*)    TERSY      87
   13 LOOP                                                              TERSY      88
      WRITE ON 6, TSATS,EQSAT                                           TERSY      89
      FORMAT(*0    TOTAL SATELLITES *,D4.1/*0    EQU. SATELLITES *,D4.2 TERSY      90
     *)                                                                 TERSY      91
      RETURN                                                            TERSY      92
      END                                                               TERSY      93
      SUBROUTINE TERMD                                                  TERMD       2
C                                                                       TERMD       3
CCC   OUTPUT STATISTICS FOR MODULES                                     TERMD       4
C                                                                       TERMD       5
      WRITE ON 6                                                        TERMD       6
      FORMAT(*1     MODULE SUMMARY*//S20,*WARN*,S24,*FAIL*,S22,*REPLACE*/ TERMD       7
     **   NAME      MIN       AVR       MAX       MIN       AVR       MAX       M TERMD       8
     *IN      AVR       MAX*/)                                         TERMD       9
      LET A = TRIG                                                      TERMD      10
      DO TO 15, FOR I=(1)(MITAB)                                        TERMD      11
      IF MNAME(I) EQ 0, GO TO 15                                        TERMD      12
      IF S121(I) + S125(I) + S129(I) EQ 0, GO TO 14                     TERMD      13
      LET B = S121(I)                                                   TERMD      14
      LET B = B/A                                                       TERMD      15
      LET C = S125(I)                                                   TERMD      16
      LET C = C/A                                                       TERMD      17
      LET D = S129(I)                                                   TERMD      18
      LET D = D/A                                                       TERMD      19
      IF N125(I) EQ 1000, LET N125(I) = 0                               TERMD      20
```

```
      IF N129(I) EQ 1000, LET N129(I) = 0                                  TERMD      21
      IF N121(I) EQ 1000, LET N121(I) = 0                                  TERMD      22
      WRITE ON 6,MNAME(I),N125(I),C,X125(I),N129(I),D,X129(I),N121(I),B,    TERMD      23
     *X121(I)                                                              TERMD      24
      FORMAT(S2,A6,I6,D7.1,2I9,D7.1,2I9,D7.1,I9)                           TERMD      25
      GO TO 15                                                             TERMD      26
   14 WRITE ON 6,MNAME(I)                                                  TERMD      27
      FORMAT(S2,A6)                                                        TERMD      28
   15 LOOP                                                                 TERMD      29
      RETURN                                                               TERMD      30
      END                                                                  TERMD      31
      ENDOGENOUS EVENT WARN                                                WARN        2
C                                                                          WARN        3
C  THIS ROUTINE WILL ATTEMPT TO SCHEDULE THE LAUNCHING OF A REPLACEMENT    WARN        4
C                                                                          WARN        5
C     MODULE.  IF SUCCESSFUL, THE CORRESPONDING FAILURE MUST BE BLOCKED    WARN        6
C                                                                          WARN        7
C     IF IT EXISTS                                                         WARN        8
C                                                                          WARN        9
      LET IEVWA = IEVWA + 1                                                WARN       10
      IF TIME GE TIMEG, LET EXMOD = MODS                                   WARN       11
      LET IS = PSAT(WARN)                                                  WARN       12
      LET IM = PMOD(WARN)                                                  WARN       13
      IF SSTAT(IS) EQ OUT, RETURN                                          WARN       14
      LET NOWAR(NOMOD(IM)) = NOWAR(NOMOD(IM)) + 1                          WARN       15
      CALL STATUS(IS,IM,8)                                                 WARN       16
      IF XSAT(IS) EQ 100, RETURN                                           WARN       17
      LET DELAY = WSATU                                                    WARN       18
      IF TIME + DELAY GT TGO(IS), RETURN                                   WARN       19
      CALL REDUN(IS,IM)                                                    WARN       20
      IF DELTA GT 0., RETURN                                               WARN       21
      CREATE QWAIT                                                         WARN       22
      LET PSAT(QWAIT) = IS                                                 WARN       23
      LET PMOD(QWAIT) = IM                                                 WARN       24
      LET TIMEA(QWAIT) = DELAY                                             WARN       25
      CAUSE QWAIT AT TIME + WAIT4                                          WARN       26
      RETURN                                                               WARN       27
      END                                                                  WARN       28
      SUBROUTINE WEIBUL (AW,BW,TW,AF,BF,TF)                                WEIBUL      2
C                                                                          WEIBUL      3
C     WEIBUL FUNCTION FOR FAILURE AND WARNING TIMES                        WEIBUL      4
C                                                                          WEIBUL      5
      LET TW = 0.                                                          WEIBUL      6
      IF AW EQ 0., GO TO 5                                                 WEIBUL      7
      IF TIMEC EQ 0., GO TO 1                                              WEIBUL      8
      LET AX = TIMEC                                                       WEIBUL      9
      GO TO 2                                                              WEIBUL     10
    1 LET AX = RANF(N)                                                     WEIBUL     11
    2 LET AX = -ALOG(AX)                                                   WEIBUL     12
      IF BW NE 1., LET AX = AX**(1./BW)                                    WEIBUL     13
      LET TW = AW*AX                                                       WEIBUL     14
      LET TF = 0.                                                          WEIBUL     15
      IF AF EQ 0., RETURN                                                  WEIBUL     16
      LET AX = TW/AF                                                       WEIBUL     17
      IF BF NE 1., LET AX = AX**BF                                         WEIBUL     18
      LET AN3 = EXP(-AX)                                                   WEIBUL     19
      IF TIMEC EQ 0., GO TO 3                                              WEIBUL     20
```

```
      LET AX = TIMEC                                                    WEIBUL     21
      GO TO 4                                                           WEIBUL     22
    3 LET AX = RANF(N)                                                  WEIBUL     23
    4 LET AX = -ALOG(AX*AN3)                                            WEIBUL     24
      IF BF NE 1., LET AX = AX**(1./BF)                                 WEIBUL     25
      LET TF = AF*AX                                                    WEIBUL     26
      RETURN                                                            WEIBUL     27
    5 LET TF = 0.                                                       WEIBUL     28
      IF AF EQ 0., RETURN                                               WEIBUL     29
      IF TIMEC EQ 0., GO TO 6                                           WEIBUL     30
      LET AX = TIMEC                                                    WEIBUL     31
      GO TO 7                                                           WEIBUL     32
    6 LET AX = RANF(N)                                                  WEIBUL     33
    7 LET AX = -ALOG(AX)                                                WEIBUL     34
      IF BF NE 1., LET AX = AX**(1./BF)                                 WEIBUL     35
      LET TF = AF*AX                                                    WEIBUL     36
      RETURN                                                            WEIBUL     37
      END                                                               WEIBUL     38
      SUBROUTINE PRFORM(DVLEG,PLEG,BOIL,NLEG,WPER,NEXIT,ERFLG)          PRFORM      3
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)              /TUGVEH/    2
     X             ,REUSE(3),WGA,TR                                     /TUGVEH/    3
     X,FEAS(2)                                                          /TUGVEH/    4
      INTEGER SPAR                                                      /TUGVEH/    5
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                       /SEPVEH/    2
     *  ,CHEM                                                          /SEPVEH/    3
     *  ,DT                                                            /SEPVEH/    4
      INTEGER SEPS                                                      /SEPVEH/    5
      REAL MS                                                           /SEPVEH/    6
      DIMENSION DVLEG(10),PLEG(10)                                      PRFORM      6
      DIMENSION DVEFF(10)                                               PRFORM      7
      INTEGER ERFLG                                                     PRFORM      8
      REAL MPLA,MPLB                                                    PRFORM      9
      IF (NSTG .LT. 0)  STOP                                            PRFORM     10
C                                                                       PRFORM     11
C                 PERF - SETS UP AND CHOOSES THE SPECIFIC               PRFORM     12
C                        PERFORMANCE SUBROUTINE TO BE EXECUTED          PRFORM     13
C                 SSHOT - SLINGSHOT - LIQUID UPPERS                     PRFORM     14
C                 SSLQD - SINGLE STAGE LIQUID                           PRFORM     15
C                 TRNKC - TRANS KICK - SOLID UPPERS                     PRFORM     16
C                 SEPSIM- SEPS SIMULATOR                                PRFORM     17
C                                                                       PRFORM     18
      IF ( SEPS .NE. 0 )  GO TO 40                                      PRFORM     19
      DO 5 I=1,NLEG                                                     PRFORM     20
    5 DVEFF(I) = DVLEG(I)*(1.+TR)                                       PRFORM     21
      IF ( NSTG .GT. 1 )  GO TO 10                                      PRFORM     22
      CALL SSLQD(DVEFF,PLEG,BOIL,NLEG)                                  PRFORM     23
      GO TO 50                                                          PRFORM     24
   10 DO 20  I = 2,NSTG                                                 PRFORM     25
      IF (SPAR(I) .NE. 0 ) GO TO 30                                     PRFORM     26
   20 CONTINUE                                                          PRFORM     27
      CALL SSHOT(DVEFF,PLEG,NLEG)                                       PRFORM     28
      GO TO 50                                                          PRFORM     29
   30 CALL TRNKC(DVEFF,PLEG)                                            PRFORM     30
      GO TO 50                                                          PRFORM     31
   40 MPLA = PLEG(1)                                                    PRFORM     32
      MPLB = PLEG(NLEG)                                                 PRFORM     33
      CALL SEPX  (MPLA, MPLB,ERFLG,NEXIT )                              PRFORM     34
```

```
      RETURN                                                            PREFORM    35
   50 WPER = 100.*(1.-AMAX1(FEAS(1),FEAS(2)))                           PREFORM    36
      RETURN                                                            PREFORM    37
      END                                                               PREFORM    38
      SUBROUTINE CONEC(NS,NVEH,ISESP)                                   CONEC       2
C                                                                       CONEC       3
CC    THIS ROUTINE WILL GET THE NECESSARY VEHICLE DATA                  CONEC       4
C                                                                       CONEC       5
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                       /SEPVEH/    2
     * ,CHEM                                                           /SEPVEH/    3
     * ,DT                                                             /SEPVEH/    4
      INTEGER SEPS                                                      /SEPVEH/    5
      REAL MS                                                           /SEPVEH/    6
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)              /TUGVEH/    2
     X             ,REUSE(3),WGA,TR                                   /TUGVEH/    3
     X,FEAS(2)                                                          /TUGVEH/    4
      INTEGER SPAR                                                      /TUGVEH/    5
      NSTG = NS                                                         CONEC       8
      CHEM = NVEH                                                       CONEC       9
      SEPS = ISESP                                                      CONEC      10
      RETURN                                                            CONEC      11
      END                                                               CONEC      12
      SUBROUTINE LINKT(I,A,B,C,D,E,JF,G,TRIN)                           LINKT       2
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)              /TUGVEH/    2
     X             ,REUSE(3),WGA,TR                                   /TUGVEH/    3
     X,FEAS(2)                                                          /TUGVEH/    4
      INTEGER SPAR                                                      /TUGVEH/    5
      EISP(I) = A                                                       LINKT       4
      WS(I) = B                                                         LINKT       5
      WPA(I) = D                                                        LINKT       6
      REUSE(I) = 1. - E                                                 LINKT       7
    5 SPAR(I) = JF                                                      LINKT       8
      WGA = G                                                           LINKT       9
      IF(I.EQ.1) TR = TRIN                                              LINKT      10
      RETURN                                                            LINKT      11
      END                                                               LINKT      12
      SUBROUTINE SSLQD(DVLEG,PLEG,BOIL,NLEG)                            SSLQD       2
C                                                                       SSLQD       3
C           SSLQD - PERFORMANCE ROUTINE FOR SINGLE STAGE LIQUID         SSLQD       4
C                                                                       SSLQD       5
C     GENERAL INPUT                                                     SSLQD       6
C          WS     THE STRUCTURE WEIGHT FOR THE STAGES                   SSLQD       7
C          WPA    THE ALLOWABLE PROPELLENT WEIGHT FOR THE STAGES        SSLQD       8
C          EISP   EFFECTIVE ISP  (SEC)                                  SSLQD       9
C          G      GRAVITY  (CONSTANT)                                   SSLQD      10
C          WGA    ALLOWABLE GROSS WEIGHT                                SSLQD      11
C          NSTG   NUMBER OF STAGES                                      SSLQD      12
C          REUSE REUSABLE FLAG  0 = EXPENDABLE , 1 = REUSABLE           SSLQD      13
C                                                                       SSLQD      14
C     SPECIFIC INPUT                                                    SSLQD      15
C          DVLEG  DELTA V FOR EACH LEG                                  SSLQD      16
C          PLEG   PAYLOAD ON EACH LEG                                   SSLQD      17
C          NLEG   NUMBER OF LEGS                                        SSLQD      18
C                                                                       SSLQD      19
C     OUTPUT                                                            SSLQD      20
C          FEAS(1)     PROPELLENT WEIGHT RATIO                        SSLQD      21
C          FEAS(2)     GROSS WEIGHT RATIO                             SSLQD      22
```

```
C                 IF  LESS THAN OR EQUAL TO 1 CONSTRAINTS NOT EXCEEDED    SSLQD      23
C                 IF  GREATER THEN 1 CONSTRAINTS EXCEEDED                 SSLQD      24
C                                                                         SSLQD      25
C                                                                         SSLQD      26
      DIMENSION DVLEG(10),PLEG(10)                                        SSLQD      27
      DIMENSION BOIL(1)                                                   SSLQD      28
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)                /TUGVEH/    2
     X             ,REUSE(3),WGA,TR                                       /TUGVEH/    3
     X,FEAS(2)                                                           /TUGVEH/    4
      INTEGER SPAR                                                       /TUGVEH/    5
      COMMON/MISC/G                                                       SSLQD      30
      REAL        MR                                                      SSLQD      31
      DATA G/32.1725/                                                     SSLQD      32
      WP = 0.0                                                            SSLQD      33
      N = NLEG                                                            SSLQD      34
      ON1=G*EISP(1)*2.                                                    SSLQD      35
      DO  10  I = 1,NLEG                                                  SSLQD      36
      EXP1 = DVLEG(N) / ON1                                               SSLQD      37
      MR =  EXP( EXP1)                                                    SSLQD      38
      WPI = (WS(1)+ WP + PLEG(N) ) * (MR - 1.0)                           SSLQD      39
      WP = WP + WPI + BOIL(N)                                             SSLQD      40
      WPI = (WS(1) + WP + PLEG(N))*(MR-1.)                                SSLQD      41
      WP =  WP +  WPI                                                     SSLQD      42
   10 N = N - 1                                                           SSLQD      43
C                                                                         SSLQD      44
C                 OK  - NOW HAVE WEIGHT FOR LEG                           SSLQD      45
C                                                                         SSLQD      46
      WG = WP + WS(1) + PLEG(1)                                           SSLQD      47
      FEAS(1) =  WP / WPA(1)                                              SSLQD      48
      FEAS(2) =  WG / WGA                                                 SSLQD      49
      RETURN                                                              SSLQD      50
      END                                                                 SSLQD      51
      SUBROUTINE  SSHOT (DVLEG,PLEG,NLEG)                                 SSHOT      2
C                                                                         SSHOT      3
C           SSHOT - PERFORMANCE ROUTINE FOR THE SLINGSHOT TYPE            SSHOT      4
C                   DEPLOYMENT - UP TO 3 LEGS AND EITHER 2 OR             SSHOT      5
C                   3 STAGES.                                             SSHOT      6
C                                                                         SSHOT      7
C       GENERAL INPUT                                                     SSHOT      8
C            WS    THE STRUCTURE WEIGHT FOR THE STAGES                    SSHOT      9
C            WPA   THE ALLOWABLE PROPELLENT WEIGHT FOR THE STAGES         SSHOT     10
C            EISP  EFFECTIVE ISP  (SEC)                                   SSHOT     11
C            G     GRAVITY  (CONSTANT)                                    SSHOT     12
C            WGA   ALLOWABLE GROSS WEIGHT                                 SSHOT     13
C            NSTG  NUMBER OF STAGES                                       SSHOT     14
C            REUSE REUSABLE FLAG  0 = EXPENDABLE , 1 = REUSABLE           SSHOT     15
C                                                                         SSHOT     16
C       SPECIFIC INPUT                                                    SSHOT     17
C            DVLEG DELTA V FOR EACH LEG                                   SSHOT     18
C            PLEG PAYLOAD ON EACH LEG                                     SSHOT     19
C            NLEG NUMBER OF LEGS                                          SSHOT     20
C                                                                         SSHOT     21
C       OUTPUT                                                            SSHOT     22
C            FEAS(1)     PROPELLENT WEIGHT RATIO                          SSHOT     23
C            FEAS(2)     GROSS WEIGHT RATIO                               SSHOT     24
C                                                                         SSHOT     25
C                 IF  LESS THAN OR EQUAL TO 1 CONSTRAINTS NOT EXCEEDED    SSHOT     26
```

```
C                IF  GREATER THEN 1 CONSTRAINTS EXCEEDED
C
      DIMENSION OVLEG(10),PLEG(10),WP(3)
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)
     X             ,REUSE(3),WGA,TR
     X,FEAS(2)
      INTEGER SPAR
      COMMON/MISC/G
      REAL        MR
C
C               INITILIZE AND COMPUTE STAGE WT
C
      WP(NSTG) = 0.0
      DN1 =EISP(NSTG) * G
      IF ( NLEG .EQ. 1 )  GO TO 20
C
C                     IF MORE THAN ONE LEG COMPUTE N WTS
C
      N = NLEG
      DO  10    I = 2,NLEG
            EX1 = OVLEG(N) / DN1
            MR  = EXP(EX1)
            WPI = (WS(NSTG) + WP(NSTG) + PLEG(N) ) * (MR -1.0)
            WP(NSTG) =  WP(NSTG) + WPI
   10 N = N - 1
C
C               COMPUTE RATIO AND TEST IF OK
C
      FEAS(1) =  WP(NSTG) / WPA(NSTG)
      FEAS(2) = 0.8
      IF ( FEAS(1) .GT. 1.0  )  RETURN
C
C                     MISSION FEASABLE - CONTINUE
C                     PICK UP SINGLE LEG COMPUTATION
C
   20 MR = (WS(NSTG) + WPA(NSTG) + PLEG(1) ) /
     X     (WS(NSTG) + WP(NSTG ) + PLEG(1) )
      T1 = ALOG( MR )
C
C               NOW FORM DELTA V FOR UPPER STAGE AND
C               SEE IF ITS SUFFICIENT
C
      DLTVU = DN1 * T1
      IF  ( DLTVU .LT. OVLEG(1) )  GO TO 30
C
C               ITS SUFFICIENT - SET FLAG AND RETURN
C
      FEAS(1) = .5
      RETURN
C
C               NO IT NEEDS MORE
C
   30 WP(NSTG) = WPA(NSTG)
      DLTVL =  OVLEG(1) - DLTVU
      DLTVLU =  0.0
      WG2    = PLEG(1)
C
```

```
SSHOT      27
SSHOT      28
SSHOT      29
/TUGVEH/    2
/TUGVEH/    3
/TUGVEH/    4
/TUGVEH/    5
SSHOT      31
SSHOT      32
SSHOT      33
SSHOT      34
SSHOT      35
SSHOT      36
SSHOT      37
SSHOT      38
SSHOT      39
SSHOT      40
SSHOT      41
SSHOT      42
SSHOT      43
SSHOT      44
SSHOT      45
SSHOT      46
SSHOT      47
SSHOT      48
SSHOT      49
SSHOT      50
SSHOT      51
SSHOT      52
SSHOT      53
SSHOT      54
SSHOT      55
SSHOT      56
SSHOT      57
SSHOT      58
SSHOT      59
SSHOT      60
SSHOT      61
SSHOT      62
SSHOT      63
SSHOT      64
SSHOT      65
SSHOT      66
SSHOT      67
SSHOT      68
SSHOT      69
SSHOT      70
SSHOT      71
SSHOT      72
SSHOT      73
SSHOT      74
SSHOT      75
SSHOT      76
SSHOT      77
SSHOT      78
SSHOT      79
SSHOT      80
```

```
C                 TEST THE NUMBER OF STAGES -                                  SSHOT      81
C                                                                              SSHOT      82
      IF ( NSTG .EQ. 2 )   GO TO 60                                            SSHOT      83
C                                                                              SSHOT      84
C                 ITS A THREE STAGE VEHICLE - SEE IF THE                       SSHOT      85
C                       SECOND STAGE IS EXPENDABLE                             SSHOT      86
      WP(2) = 0.0                                                              SSHOT      87
      IF(IFIX(REUSE(2)).EQ.0) GO TO 40                                         SSHOT      88
      EXP2 =  DLTVL*REUSE(2) /(G*EISP(2))                                      SSHOT      89
      MR =  EXP (EXP2)                                                         SSHOT      90
      WP(2) = WS(2) * ( MR - 1.0 )                                             SSHOT      91
C                                                                              SSHOT      92
C                 TEST IF THERE IS ENOUGH PROPELLENT                           SSHOT      93
C                                                                              SSHOT      94
      IF  ( WP(2) .LT. WPA(2) )  GO TO 40                                      SSHOT      95
C                                                                              SSHOT      96
C                 NO - SECOND STAGE CANNOT EVEN RETURN - ABORT                 SSHOT      97
C                                                                              SSHOT      98
      FEAS(1) = 1.5                                                            SSHOT      99
      RETURN                                                                   SSHOT     100
C                                                                              SSHOT     101
C                 ITS OK - CONTINUE                                            SSHOT     102
C                                                                              SSHOT     103
   40 WG2 = PLEG(1) + WP(3) + WS(3)                                            SSHOT     104
      MR =  (WS(2) + WPA(2) + WG2 ) /  (WS(2) + WP(2) + WG2 )                  SSHOT     105
      DLTVLU = G* EISP(2) * ALOG(MR)                                           SSHOT     106
C                                                                              SSHOT     107
C                 TEST IF SECOND STAGE CAN DO THE MISSION                      SSHOT     108
C                                                                              SSHOT     109
      IF ( DLTVLU .LT. DLTVL )   GO TO  50                                     SSHOT     110
      FEAS(1) = .7                                                             SSHOT     111
      RETURN                                                                   SSHOT     112
C                                                                              SSHOT     113
C                 NO CONTINUE                                                  SSHOT     114
C                                                                              SSHOT     115
   50 WP(2) = WPA(2)                                                           SSHOT     116
C                                                                              SSHOT     117
C                       ONLY TWO STAGE RETURN                                  SSHOT     118
C                                                                              SSHOT     119
   60 DLTVLL = DLTVL - DLTVLU                                                  SSHOT     120
      T2 = G * EISP(1)                                                         SSHOT     121
C                                                                              SSHOT     122
C                 SET UP AND TEST IF THE STAGE IS REUSABLE                     SSHOT     123
C                                                                              SSHOT     124
      WP(1) = 0.0                                                              SSHOT     125
      IF(IFIX(REUSE(1)).EQ.0) GO TO 70                                         SSHOT     126
C                                                                              SSHOT     127
C                 NO COMPUTE THE WP                                            SSHOT     128
C                                                                              SSHOT     129
      EXP3 =  DLTVLL*REUSE(1)  /T2                                             SSHOT     130
      MR = EXP (EXP3)                                                          SSHOT     131
      WP(1) = WS(1) * (MR -1.0)                                                SSHOT     132
C                                                                              SSHOT     133
C                 TEST IF FIRST STAGE CAN RETURN                               SSHOT     134
C                                                                              SSHOT     135
      IF ( WP(1) .LT.  WPA(1) )  GO TO 70                                      SSHOT     136
      FEAS(1) = 1.3                                                            SSHOT     137
```

```
      RETURN                                                            SSHOT     138
C                                                                       SSHOT     139
CC                  ITS OK - CONTINUE                                   SSHOT     140
C                                                                       SSHOT     141
   70 EXP4 = DLTVLL/T2                                                  SSHOT     142
      MR = EXP (EXP4)                                                   SSHOT     143
      WG2 = WG2 + WP(2) + WS(2)                                         SSHOT     144
      WPI = (WS(1) + WP(1) + WG2) *(MR-1.0)                             SSHOT     145
      WP1     = WP(1) + WPI                                             SSHOT     146
      FEAS(1) = WP1    / WPA(1)                                         SSHOT     147
      WG = WG2 + WS(1) + WP1                                            SSHOT     148
      FEAS(2) = WG / WGA                                                SSHOT     149
      RETURN                                                            SSHOT     150
      END                                                               SSHOT     151
      SUBROUTINE TRNKC(DVLEGX,PLEG)                                     TRNKC       2
C                                                                       TRNKC       3
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCTRNKC       4
CC          GENERAL INPUT                                             TRNKC       5
CC                WS     THE STRUCTURE WEIGHT FOR THE STAGES          TRNKC       6
CC                WPA    THE ALLOWABLE PROPELLENT WEIGHT FOR THE STAGES TRNKC      7
CC                EISP   EFFECTIVE ISP  (SEC)                         TRNKC       8
CC                G      GRAVITY  (CONSTANT)                          TRNKC       9
CC                WGA    ALLOWABLE GROSS WEIGHT                       TRNKC      10
CC                NSTG   NUMBER OF STAGES                             TRNKC      11
CC                REUSE  REUSABLE FLAG  0 = EXPENDABLE , 1 = REUSABLE TRNKC      12
CC                                                                    TRNKC      13
CC          SPECIFIC INPUT                                            TRNKC      14
CC                DVLEG(1)  DELTA V FOR LOW ALTITUDE BURN             TRNKC      15
CC                DVLEG(2)  DELTA V FOR HIGH ALTITUDE BURN            TRNKC      16
CC                NLEG      SET EQUAL TO 2                            TRNKC      17
CC                                                                    TRNKC      18
CC          OUTPUT                                                    TRNKC      19
CC                FEAS(1)     PROPELLENT WEIGHT RATIO                 TRNKC      20
CC                FEAS(2)     GROSS WEIGHT RATIO                      TRNKC      21
CC                                                                    TRNKC      22
CC                IF  LESS THAN OR EQUAL TO 1 CONSTRAINTS NOT EXCEEDED TRNKC     23
CC                IF  GREATER THEN 1 CONSTRAINTS EXCEEDED             TRNKC      24
C                                                                     TRNKC      25
      DIMENSION PLEG(1),WP(3)                                         /TUGVEH/   2
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)            /TUGVEH/   3
     X             ,REUSE(3),WGA,TR                                   /TUGVEH/   4
     X,FEAS(2)                                                        /TUGVEH/   5
      INTEGER SPAR                                                    TRNKC      27
      COMMON/MISC/G                                                   TRNKC      28
      COMMON/DELTAV/DVLEG(2)                                          TRNKC      29
C                                                                     TRNKC      30
CCC                 INITILIZE AND COMPUTE STAGE WT                    TRNKC      31
C                                                                     TRNKC      32
      REAL          MRK2,MRKMX,MRCK,MRAB,MR1                          TRNKC      33
      FEAS(1) = 0.5                                                   TRNKC      34
      FEAS(2) = 0.5                                                   TRNKC      35
      IF ( NSTG .EQ. 2 )  GO TO 10                                    TRNKC      36
      WPL2 = PLEG(1)                                                  TRNKC      37
      DVK2 = DVLEG(2)                                                 TRNKC      38
      EXP1 =  DVK2 / (G * EISP(3) )                                   TRNKC      39
      MRK2 =  EXP (EXP1)                                              TRNKC      40
C                                                                     TRNKC      41
CC                  SECOND KICK MUST DO ALL OF SECOND BURN
```

```
C                NOW GET FUEL REQUIRED FOR SECOND KICK                  TRNKC      42
C                                                                       TRNKC      43
      WP(3) = (WS(3) + WPL2) * (MRK2 -1.0)                              TRNKC      44
      FEAS(1) =  WP(3) / WPA(3)                                         TRNKC      45
C                                                                       TRNKC      46
CC               IF THE SECOND KICK CANNOT DO THE SECOND BURN, EXIT.    TRNKC      47
C                                                                       TRNKC      48
      IF  ( FEAS(1) .GT.  1.0)  RETURN                                  TRNKC      49
C                                                                       TRNKC      50
CCC              SECOND STAGE ASSUMED FULL - EXCESS FUEL IS             TRNKC      51
CCC              USED UP BY YAW STEERING                                TRNKC      52
C                                                                       TRNKC      53
      WGK2 = WS(3) + WPL2 + WPA(3)                                      TRNKC      54
C                                                                       TRNKC      55
CCC              NOW SET UP PARAMETERS FOR FIRST KICK                   TRNKC      56
C                                                                       TRNKC      57
      DVC =  DVLEG(1)*.75                                               TRNKC      58
      DVAB = DVLEG(1) -DVC                                              TRNKC      59
      WPL = WGK2                                                        TRNKC      60
      GO TO 20                                                          TRNKC      61
C                                                                       TRNKC      62
CC               MUST SET UP PARAMETERS IF THERE WAS NO SECOND KICK     TRNKC      63
C                                                                       TRNKC      64
   10 DVC = DVLEG (2)                                                   TRNKC      65
      DVAB = DVLEG(1)                                                   TRNKC      66
      WPL = PLEG(1)                                                     TRNKC      67
C                                                                       TRNKC      68
CC               NOW CONTINUE THE PROCESS                               TRNKC      69
C                                                                       TRNKC      70
   20 MRKMX =  1.0 + ( WPA(2) / (WPL + WS(2) ) )                        TRNKC      71
      EXP2 =  DVC / (G* EISP(2))                                        TRNKC      72
      MRCK =  EXP (EXP2)                                                TRNKC      73
      EXP3 =  DVAB/ (G* EISP(1))                                        TRNKC      74
      MRAB = EXP (EXP3)                                                 TRNKC      75
C                                                                       TRNKC      76
CC               CHECK IF KICK HAS MORE FUEL THEN REQUIRED              TRNKC      77
C                                                                       TRNKC      78
      REXP  =REUSE(1) +1.0                                              TRNKC      79
      IF ( MRKMX .GT. MRCK )    MRKMX = MRCK                            TRNKC      80
      ALFINV =  EISP(2) / EISP(1)                                       TRNKC      81
      MR1 = MRAB * MRCK ** ALFINV / MRKMX**ALFINV                       TRNKC      82
      WP(2)=(WS(2) + WPL ) * (MRKMX -1.0 )                              TRNKC      83
      WP(1) = ((MR1-1.)*(WPA(2)+WS(2)+WPL)) + ((MR1**REXP-1.)           TRNKC      84
     *        * WS(1) )                                                  TRNKC      85
      WG =  WPL + WPA(2) +WS(2) + WP(1) + WS(1)                         TRNKC      86
      FEAS(1) =  WP(1) / WPA(1)                                         TRNKC      87
      FEAS(2) =  WG / WGA                                               TRNKC      88
      RETURN                                                            TRNKC      89
      END                                                               TRNKC      90
      SUBROUTINE LDSEP(A,B,C,D,H,I,F,G)                                 LDSEP       2
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                       /SEPVEH/    2
     *  ,THEM                                                           /SEPVEH/    3
     *  ,DT                                                             /SEPVEH/    4
      INTEGER SEPS                                                      /SEPVEH/    5
      REAL MS                                                           /SEPVEH/    6
      COMMON /OUTP/ TD,TU,HCO,TCOS,MDT                                  /OUTP/      2
     *  ,TLEFT(5),MPT(5),TSAVE(5),RTCAP(5),MPTSV(5)                     /OUTP/      3
```

```
      REAL MDT,MPT,MPTSV,ICOS                                                   /QUIP/      4
      REAL I                                                                    LDSEP       5
      MS = A                                                                    LDSEP       6
      E  = B                                                                    LDSEP       7
      P  = C                                                                    LDSEP       8
      SISP = D                                                                  LDSEP       9
      SR = H                                                                    LDSEP      10
      SEPK = 1. - I                                                             LDSEP      11
      TSEP = F                                                                  LDSEP      12
      MDT = G                                                                   LDSEP      13
      RETURN                                                                    LDSEP      14
      END                                                                       LDSEP      15
      SUBROUTINE SEPSV(N,PER,VS,DT,PAY)                                         SEPSV       2
      DIMENSION DT(10),PAY(10)                                                  SEPSV       3
      COMMON/SERVIS/NSERV,DTHETA(10),MPLS(10),PSERV,VSERV                       /SERVIS/    4
      REAL MPLS                                                                 /SERVIS/    5
      NSERV = N                                                                 SEPSV       6
      DO 5 I = 1,NSERV                                                          SEPSV       7
      DTHETA(I) = DT(I)                                                         SEPSV       8
    5 MPLS(I) = PAY(I)                                                          SEPSV       9
      RETURN                                                                    SEPSV      10
      END                                                                       TWOBR       2
      SUBROUTINE TWOBR(DV,DV1)                                                  TWOBR       3
C                                                                               TWOBR       4
C   TRANSFER ON TWO DV S RATHER THAN ONE                                        TWOBR       5
C                                                                               TWOBR       6
      COMMON/DELTAV/DVLEG(2)                                                    TWOBR       7
      DVLEG(1) = DV1*1.01                                                       TWOBR       8
      DVLEG(2) = (DV - DV1)*1.01                                                TWOBR       9
      RETURN                                                                    TWOBR      10
      END                                                                       SEPX        2
      SUBROUTINE  SEPX   (MPLA,MPLB,ERFLG,NEXIT )                               SEPX        3
C                                                                               SEPX        4
C                 SEPX  THE SEP EXECTIVE ROUTINE IT PERFORMS THE                SEPX        5
C                       LOGIC OF UTILIZING OF THE SEPS VEHICLE                  SEPX        6
C                                                                               SEPX        7
C                 SPECFIC INPUT                                                 SEPX        8
C                       MPLA   PAYLOAD TO BE DEPLOYED                           SEPX        9
C                       MPLB   PAYLOAD TO BE RETRIEVED                          SEPX       10
C                       ERFLG   0 = DO NOT ERASE PREVIOUS MANEUVER              SEPX       11
C                               1 = ERASE THE PREVIOUS MANEUVER                 SEPX       12
C                       NEXIT SET TO 0 ON DATA CARD OF DRIVER                   SEPX       13
C                                                                               SEPX       14
C                 COMMON INPUT (SEPVEH)                                         SEPX       15
C                       MS                                                      SEPX       16
C                       MPT    AMOUNT OF FUEL REMAINING                         SEPX       17
C                       TLEFT  AMOUNT OF TIME REMAINING                         SEPX       18
C                       E                                                       SEPX       19
C                       P                                                       SEPX       20
C                       SISP   SPECIFIC IMPULSE SEPS                            SEPX       21
C                       MDT                                                     SEPX       22
C                       RTCAP                                                   SEPX       23
C                       TSEP                                                    SEPX       24
C                       RSEP                                                    SEPX       25
C                       SS     GRAVITY CONSTANT                                 SEPX       26
C                                                                               SEPX       27
C                 OUTPUT
```

```
C                     NEXIT TYPE OF EXIT FROM SEPIM SUBROUTINE                  SEPX      28
C                     NTUGS NUMBER OF TUG FLIGHTS REQUIRED TO DO THE            SEPX      29
C                           MISSION AND RETURN THE EXPENDED SEPS,IF             SEPX      30
C                           ANY.  NTUGS WILL BE BETWEEN 1 AND 3.                SEPX      31
C                     TLEFT TIME AND FUEL REMAINING ON SEPS VEHEHICLE           SEPX      32
C                     MPT         IN ORBIT                                      SEPX      33
C                                                                               SEPX      34
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)                      /TUGVEH/   2
     X             ,REUSE(3),WGA,TR                                           /TUGVEH/   3
     X,FEAS(2)                                                                 /TUGVEH/   4
      INTEGER SPAR                                                             /TUGVEH/   5
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                              /SEPVEH/   2
     *  ,CHEM                                                                  /SEPVEH/   3
     *  ,DT                                                                    /SEPVEH/   4
      INTEGER SEPS                                                             /SEPVEH/   5
      REAL MS                                                                  /SEPVEH/   6
      COMMON/SERVIS/NSERV,OTHETA(10),MPLS(10),PSERV,VSERV                      /SERVIS/   2
      REAL MPLS                                                                /SERVIS/   3
      COMMON /OUTP/ TD,TU,HCO,ICOS,MDT                                         /OUTP/     2
     *  ,TLEFT(5),MPT(5),TSAVE(5),RTCAP(5),MPTSV(5)                            /OUTP/     3
      REAL MDT,MPT,MPTSV,ICOS                                                  /OUTP/     4
      COMMON/C2/TS                                                             SEPX      39
      REAL MPLA,MPLB                                                           SEPX      40
      INTEGER     ERFLG                                                        SEPX      41
      HCO=160.                                                                 SEPX      42
      TU=0.0                                                                   SEPX      43
      TS=0.0                                                                   SEPX      44
      TD=0.0                                                                   SEPX      45
      ICOS=28.5                                                                SEPX      46
C                                                                              SEPX      47
      IF (NEXIT .GE. 1 )  GO TO 10                                             SEPX      48
C                                                                              SEPX      49
CC                  INITIALIZATION CALCULATATIONS                              SEPX      50
C                                                                              SEPX      51
      C = SISP * 9.80621                                                       SEPX      52
      DT=(E*P*4.409246)/(C*C)                                                  SEPX      53
      MPT(SEPS)=MDT                                                            SEPX      54
      TSEP = MPT(SEPS)/(86400.0*DT)                                            SEPX      55
      TLEFT(SEPS) = TSEP                                                       SEPX      56
      TSAVE(SEPS) = TSEP                                                       SEPX      57
      MPTSV(SEPS) = MPT(SEPS)                                                  SEPX      58
      RTCAP(SEPS)=0                                                            SEPX      59
C                       INITIALIZATION COMPLETE                                SEPX      60
CC                                                                             SEPX      61
C                                                                              SEPX      62
10    CONTINUE                                                                 SEPX      63
      IF ( ERFLG .GE. 1 )  GO TO 20                                            SEPX      64
C                                                                              SEPX      65
CC                      NO - SAVE PRESENT CONDITIONS                           SEPX      66
C                                                                              SEPX      67
      MPTSV(SEPS) = MPT(SEPS)                                                  SEPX      68
      TSAVE(SEPS) = TLEFT(SEPS)                                                SEPX      69
      GO TO 30                                                                 SEPX      70
C                                                                              SEPX      71
CC                      ERASE -                                                SEPX      72
C                                                                              SEPX      73
20    TLEFT(SEPS) = TSAVE(SEPS)                                                SEPX      74
```

```
      MPT(SEPS) = MPTSV(SEPS)                                                SEPX      75
30    CONTINUE                                                               SEPX      76
C                                                                            SEPX      77
CCCC                    NOW TRY TO PERFORM THE REMAINING MISSION             SEPX      78
CCCC                    WITH THE PRESENT SEPS                                SEPX      79
C                                                                            SEPX      80
   40 CALL  SEPIM (MPLA,MPLB,0,NEXIT )                                       SEPX      81
C                                                                            SEPX      82
CCC                 SEE IF IT CAN BE DONE - 1,2,5,6 OK -  3,4,7  NO          SEPX      83
C                                                                            SEPX      84
      IF(MPT(SEPS).GT.RTCAP(SEPS)) RETURN                                    SEPX      85
      IF(NEXIT.EQ.3) RETURN                                                  SEPX      86
      IF(NEXIT.EQ.4) RETURN                                                  SEPX      87
      IF(NEXIT.EQ.7) RETURN                                                  SEPX      88
      IF(NEXIT.EQ.8) RETURN                                                  SEPX      89
      IF(NEXIT.EQ.9) RETURN                                                  SEPX      90
      IF(NEXIT.EQ.10) RETURN                                                 SEPX      91
      NEXIT=NEXIT + 2                                                        SEPX      92
      IF(NEXIT.EQ.8) NEXIT = 9                                               SEPX      93
      RETURN                                                                 SEPX      94
      END                                                                    SEPX      95
      SUBROUTINE  FAZS                                                       FAZS       2
C                                                                            FAZS       3
CCC   PERFORMS SEPS PHASING, ASSUMING CONSTANT SEP THRUSTING.                FAZS       4
CCC   INPUTS:  NSERV=NUMBER OF SERVICE LEGS.                                 FAZS       5
CCC            DTHETA= ANGULAR TRAVEL (DEG) OF EACH SERVICE LEG.             FAZS       6
CCC            MPLS= PAYLOAD (LBS) ON EACH SERVICE LEG.                      FAZS       7
CCC            PSERV,VSERV= PERIOD (SEC) AND VELOCITY (MPS) OF SERVICE ORBI  FAZS       8
CCC   OUTPUTS:  MPT= FEUL REMAINIG AFTER PHASING (LBS).                      FAZS       9
CCC             TLEFT= TIME REMAINING ON SEPS AFTER PHASING (DAYS).          FAZS      10
C                                                                            FAZS      11
      REAL MKG                                                               FAZS      12
      COMMON /OUTP/ TD,TU,HCO,ICOS,MDT                                       /OUTP/     2
     * ,TLEFT(5),MPT(5),TSAVE(5),RTCAP(5),MPTSV(5)                          /OUTP/     3
      REAL MDT,MPT,MPTSV,ICOS                                                /OUTP/     4
      COMMON/SERVIS/NSERV,DTHETA(10),MPLS(10),PSERV,VSERV                    /SERVIS/   2
      REAL MPLS                                                              /SERVIS/   3
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                            /SEPVEH/   2
     * ,CHEM                                                                 /SEPVEH/   3
     * ,DT                                                                   /SEPVEH/   4
      INTEGER SEPS                                                           /SEPVEH/   5
      REAL MS                                                                /SEPVEH/   6
      COMMON/C2/TS                                                           FAZS      16
      COMMON/TSA/TPLS(30),TUP,TDOWN                                          FAZS      17
      DATA PSERV,VSERV/86165.,3074.66/                                       FAZS      18
      IF(CHEM.NE.0) RETURN                                                   FAZS      19
      F=(DT*9.80621*SISP)/2.204623                                           FAZS      20
      CONST1=(3.0*F*PSERV)/(4.0*VSERV)                                       FAZS      21
      TS = TLEFT(SEPS)                                                       FAZS      22
      DO 100 N=1,NSERV                                                       FAZS      23
      MKG = (MS+MPT(SEPS)+MPLS(N))/2.204623                                  FAZS      24
      REV=SQRT((MKG*ABS(DTHETA(N)))/(360.*CONST1))                           FAZS      25
      TLEFT(SEPS)=TLEFT(SEPS)-((REV*PSERV)/86400.)                           FAZS      26
      TPLS(N) = TS - TLEFT(SEPS)                                             FAZS      27
      MPT(SEPS) = MPT(SEPS) -DT*REV*PSERV                                    FAZS      28
  100 CONTINUE                                                               FAZS      29
      TS = TS - TLEFT(SEPS)                                                  FAZS      30
```

```
      RETURN                                                            EAZS       31
      END                                                               EAZS       32
      SUBROUTINE TPHAS(A,N)                                             TPHAS       2
      COMMON/TSA/TPLS(30),TUP,TDOWN                                     TPHAS       3
      DIMENSION A(1)                                                    TPHAS       4
      COMMON /OUTP/ TD,TU,HCO,ICOS,MDT                                  /OUTP/      2
     * ,TLEFT(5),MPT(5),TSAVE(5),RTCAP(5),MPTSV(5)                      /OUTP/      3
      REAL MDT,MPT,MPTSV,ICOS                                           /OUTP/      4
      A(1) = TU/360.                                                    TPHAS       6
      A(N) = TD/360.                                                    TPHAS       7
      IF(N.EQ.2) RETURN                                                 TPHAS       8
      NX = N-2                                                          TPHAS       9
      DO 5 I=1,NX                                                       TPHAS      10
    5 A(I+1) = TPLS(I)/360.                                             TPHAS      11
      RETURN                                                            TPHAS      12
      END                                                               TPHAS      13
      SUBROUTINE  SEPIM (MPLA,MPLB,KSEP,NEXIT)                          SEPIM       2
C                                                                       SEPIM       3
C             SEPIM THIS SUBROUTINE COMPUTES THE PERFORMANCE            SEPIM       4
C                   OF THE SEPS ON A DEPLOY MISSION.                    SEPIM       5
C                                                                       SEPIM       6
C             SPECIFIC INPUT                                            SEPIM       7
C                   MPLA   PAYLOAD TO BE DEPLOYED                       SEPIM       8
C                   MPLB   PAYLOAD TO BE RETRIEVED                      SEPIM       9
C                   KSEP   ERASE FLAG                                   SEPIM      10
C                               0 = DONT ERASE PRIEVIOUS MANEUVER       SEPIM      11
C                               1 = ERASE PPIEVIOUS MANEUVER            SEPIM      12
C                   NEXIT SET TO 0 PRIOR TO ENTRY                       SEPIM      13
C                                                                       SEPIM      14
C             OUTPUT                                                    SEPIM      15
C                   NEXIT TYPE OF EXIT FROM SEPS IF MISSION POSSIBLE    SEPIM      16
C                   NTUGS NUMBER OF TUG FLIGHTS REQUIRED TO             SEPIM      17
C                         DO THE MISSION AND RETURN EXPENDED SEPS,      SEPIM      18
C                         IF ANY.  NTUGS WILL BE BETWEEN 1 AND 3        SEPIM      19
C                   TLFT   TIME AND FUEL REMAINING ON SEPS IN ORBIT     SEPIM      20
C                   MPT                                                 SEPIM      21
C                                                                       SEPIM      22
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                       /SEPVEH/    2
     * ,CHEM                                                            /SEPVEH/    3
     * ,DT                                                              /SEPVEH/    4
      INTEGER SEPS                                                      /SEPVEH/    5
      REAL MS                                                           /SEPVEH/    6
      COMMON /OUTP/ TD,TU,HCO,ICOS,MDT                                  /OUTP/      2
     * ,TLEFT(5),MPT(5),TSAVE(5),RTCAP(5),MPTSV(5)                      /OUTP/      3
      REAL MDT,MPT,MPTSV,ICOS                                           /OUTP/      4
      COMMON/SERVIS/NSERV,DTHETA(10),MPLS(10),PSERV,VSERV               /SERVIS/    2
      REAL MPLS                                                         /SERVIS/    3
      COMMON/TABLE/TUGDV(20)                                            SEPIM      26
      REAL MPLA,MPLB,MRTUG                                              SEPIM      27
      TU=0.0                                                            SEPIM      28
      TD=0.0                                                            SEPIM      29
      HCO=160.                                                          SEPIM      30
      ICOS=28.5                                                         SEPIM      31
C                                                                       SEPIM      32
C             FIRST TEST IF THERE IS A SEPS AVAILABLE                   SEPIM      33
C                                                                       SEPIM      34
      IF(TLEFT(SEPS).LT.TSEP-.001) GO TO 20                             SEPIM      35
```

```
      CHEM = 0
      PUT IN WEIGHT CONSTRAINED TUG
C
C                 NO - ITS A NEW SEPS
      TLEFT(SEPS) =TSEP
      WPLA = MPLA + MS + MPT(SEPS)
      WPLB =  0.0
C
CC    FIND PROPELLANT REQUIRED TO RETRIEVE SEPS
C
      RTCAP(SEPS) = 0
      IF(SEPK.EQ.0) GO TO 9
      CALL TUGCP(0,MS,MRTUG,DVTUG)
      IF(DVTUG.GE.TUGDV(13)) GO TO 9
      IF(DVTUG.LT.TUGDV(1)) GO TO 10
      X=MPT(SEPS)
      MPT(SEPS) = 0.
      CALL INTORB(DVTUG,HCO,ICOS)
      CALL SEPDV(HCO,ICOS,DVSEP,MRSEP)
5     RTCAP(SEPS) = MPT(SEPS)
      CALL PLUPD(0.,MRSEP,TD)
      IF(MPT(SEPS).GT.0.) GO TO 8
      MPT(SEPS) = -MPT(SEPS) + RTCAP(SEPS) + 5.
      GO TO 5
8     CONTINUE
      MPT(SEPS) = X
      TLEFT(SEPS) = TSEP
9     CONTINUE
CCC
CC                CALL TUGCP TO DETERMINE TUG CAPABILITY
C
      CALL TUGCP  (WPLA,WPLB,MRTUG,DVTUG)
      IF (DVTUG .LT. TUGDV(13) )  GO TO 10
C
CCC                     TUG DELIVERS SEPS AND MPLA TO SYNC EQ:
C
      TLEFT(SEPS) = TLEFT(SEPS) - .005
      TU = 0.0
      TD = 0.0
      HCO=19323.
      ICOS=0.0
      IF (NSERV.GT.0) CALL FAZS
      NEXIT = 2
      RETURN
C
CCC                     NEXT CHECK IF ITS CAPABLE
C
   10 NEXIT = 3
      IF ( DVTUG .LT. TUGDV(1) )   RETURN
C
CCC               ITS OK - CONTINUE DETERMINE CHANEOVER ORBIT
C
      CALL  INTORB (DVTUG,HCO,ICOS)
C
CC                DETERMINE THE SEPS DELTA V
C
      CALL  SEPDV (HCO,ICOS,DVSEP,MRSEP)
```

SEPIM 36 through SEPIM 92

```
C
C                PERFORM UP LEG AND PHASING                                    SEPIM      93
C                                                                              SEPIM      94
C                                                                              SEPIM      95
      CALL  PLUPD (MPLA,MRSEP,TU )                                             SEPIM      96
      IF (NSERV.GT.0) CALL FAZS                                                SEPIM      97
C                                                                              SEPIM      98
C                SET NEXIT AND TEST IF THERE IS FUEL REMAINING                 SEPIM      99
C                                                                              SEPIM     100
      NEXIT = 1                                                                SEPIM     101
      IF(MPT(SEPS) .GE.0.) RETURN                                              SEPIM     102
C                                                                              SEPIM     103
C                SEPS CANNOT DELIVER THE PAYLOAD - SET FLAG AND ABORT          SEPIM     104
C                                                                              SEPIM     105
      NEXIT = 4                                                                SEPIM     106
      RETURN                                                                   SEPIM     107
C                                                                              SEPIM     108
C                THIS ENTRY POINT FOR SEPS AVAILABLE                           SEPIM     109
C                IN SYNC EQ. ORBIT                                             SEPIM     110
C                                                                              SEPIM     111
   20 A = KSEP                                                                 SEPIM     112
      SMPT = MPT(SEPS)                                                         SEPIM     113
      WPLB = MPLB+A*SEPK*NS                                                    SEPIM     114
      WPLA = MPLA                                                              SEPIM     115
C                                                                              SEPIM     116
C                     DETERMINE THE TUG CAPABILITY                             SEPIM     117
C                                                                              SEPIM     118
      CALL  TUGCP (WPLA,WPLB,MRTUG,DVTUG )                                     SEPIM     119
      IF ( DVTUG .LT. TUGDV(13) )  GO TO 30                                    SEPIM     120
C                                                                              SEPIM     121
C                     NO - TUG ALONE CAN DELIVER AND RETRIEVE                  SEPIM     122
C                          PAYLOADS TO SYNC EQ ORBIT                           SEPIM     123
C                                                                              SEPIM     124
      TU = 0.0                                                                 SEPIM     125
      TD = 0.0                                                                 SEPIM     126
      HCO=19323.                                                               SEPIM     127
      ICOS=0.0                                                                 SEPIM     128
      IF (NSERV.GT.0) CALL FAZS                                                SEPIM     129
      NEXIT = 9                                                                SEPIM     130
      IF(MPT(SEPS).LT.0.) RETURN                                               SEPIM     131
      NEXIT = 6                                                                SEPIM     132
      RETURN                                                                   SEPIM     133
C                                                                              SEPIM     134
C                     TUG ALONE CAN NOT DO IT- CHECK IF ALL OK                 SEPIM     135
C                                                                              SEPIM     136
   30 NEXIT = 7                                                                SEPIM     137
      IF ( DVTUG .LT. TUGDV(1) ) RETURN                                        SEPIM     138
C                                                                              SEPIM     139
C                ITS OK - CONTINUE                                             SEPIM     140
C                     DETERMINE CHANGEOVER ORBIT                               SEPIM     141
C                                                                              SEPIM     142
      CALL  INTORB (DVTUG,HCO,ICOS )                                           SEPIM     143
      CALL  SEPDV (HCO,ICOS,DVSEP,MRSEP)                                       SEPIM     144
      CALL  PLUPD (MPLB,MRSEP,TD )                                             SEPIM     145
C                                                                              SEPIM     146
C                     SET UP AND CHECK CONSTRAINTS                             SEPIM     147
C                                                                              SEPIM     148
      NEXIT = 8                                                                SEPIM     149
```

```
      IF(MPT(SEPS).LT. 0.) GO TO 50                                     SEPIM     150
      IF ( KSEP .EQ. 0 )   GO TO 40                                     SEPIM     151
C                                                                       SEPIM     152
CCC                        SEPS RETRIEVED ALONG WITH PAYLOAD            SEPIM     153
C                                                                       SEPIM     154
      TU = 0.0                                                          SEPIM     155
      NEXIT = 10                                                        SEPIM     156
      RETURN                                                            SEPIM     157
C                                                                       SEPIM     158
CCC                        CONTINUE PROCESS                             SEPIM     159
C                                                                       SEPIM     160
   40 CALL  PLUPD (MPLA,MRSEP,TU)                                       SEPIM     161
      IF (NSERV.GT.0) CALL FAZS                                         SEPIM     162
      NEXIT = 9                                                         SEPIM     163
      IF(MPT(SEPS).LT.0.) GO TO 60                                      SEPIM     164
C                                                                       SEPIM     165
CCC                        MISSION COMPLETE                             SEPIM     166
C                                                                       SEPIM     167
      NEXIT = 5                                                         SEPIM     168
      RETURN                                                            SEPIM     169
C                                                                       SEPIM     170
CCCC  FIND DOWN FLIGHT TIME NEXIT = 8                                   SEPIM     171
C                                                                       SEPIM     172
   50 TU=0                                                              SEPIM     173
      MPT(SEPS) = SMPT                                                  SEPIM     174
      CALL PLUPD(0.,MRSEP,TD)                                           SEPIM     175
      RETURN                                                            SEPIM     176
C                                                                       SEPIM     177
CCCC  FIND DOWN FLIGHT TIME NEXIT = 9                                   SEPIM     178
C                                                                       SEPIM     179
   60 TU = 0                                                            SEPIM     180
      MPT(SEPS) = SMPT                                                  SEPIM     181
      CALL PLUPD(MPLB,MRSEP,TD)                                         SEPIM     182
      RETURN                                                            SEPIM     183
      END                                                               SEPIM     184
      SUBROUTINE  TUGCP (WPLA,WPLB,MRTUG,DVTUG )                        TUGCP       2
C                                                                       TUGCP       3
CCCCC               TUGCP - CALLS THE APPROPRIATE TUG EQUATIONS.        TUGCP       4
                          ( AT PRESENT - ONLY OPTION IS SINGLE          TUGCP       5
                            STAGE CRYOGENIC TUG. )                      TUGCP       6
C                                                                       TUGCP       7
      CALL  CRY01 (WPLA,WPLB,MRTUG,DVTUG)                               TUGCP       8
      RETURN                                                            TUGCP       9
      END                                                               TUGCP      10
      SUBROUTINE GETFR(FR,LL,IK)                                        GETFR       2
      DIMENSION FR(4)                                                   GETFR       3
      L = LL + 9                                                        GETFR       4
      IK = 0                                                            GETFR       5
      READ(L) FR                                                        GETFR       6
      IF(EOF(L)) 20,10                                                  GETFR       7
   10 RETURN                                                            GETFR       8
   20 IK = 1                                                            GETFR       9
      RETURN                                                            GETFR      10
      END                                                               GETFR      11
      SUBROUTINE PUTFR(FR,LL,IK)                                        PUTFR       2
      DIMENSION FR(4)                                                   PUTFR       3
      L = LL + 9                                                        PUTFR       4
```

```
      IF(IK.EQ.1) GO TO 5                                                PUTFR      5
      WRITE(L) FR                                                        PUTFR      6
      RETURN                                                             PUTFR      7
    5 ENDFILE L                                                          PUTFR      8
      REWIND L                                                           PUTFR      9
      RETURN                                                             PUTFR     10
      END                                                                PUTFR     11
      SUBROUTINE  CRYO1 (WPLA,WPLB,MRTUG,DVTUG)                          CRYO1      2
C                                                                        CRYO1      3
C               CRYO1- FINDS THE DELTA V CAPABILITY OF A                 CRYO1      4
C                      SINGLE STAGE TUG WITH PAYLOADS WPLA AND WPLB.     CRYO1      5
C                                                                        CRYO1      6
C                                                                        CRYO1      7
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)               /TUGVEH/   2
     X             ,REUSE(3),WGA,TR                                      /TUGVEH/   3
     X,FEAS(2)                                                          /TUGVEH/   4
      INTEGER SPAR                                                       /TUGVEH/   5
      COMMON/MISC/G                                                      CRYO1      9
      REAL        MRTUG                                                  CRYO1     10
      WP=WPA(1)                                                          CRYO1     11
      IF ((WS(1)+WPA(1)+WPLA) .GT.WGA) WP=WGA-(WS(1)+WPLA)               CRYO1     12
      MRTUG=(WP+WS(1)+WPLA)/(WS(1)+WPLA)                                 CRYO1     13
      IF (REUSE(1).LT.0.5) GO TO 100                                     CRYO1     14
      BZ=WS(1)+WS(1)+WPLA+WPLB                                           CRYO1     15
      CZ=-WP*(WPLB+WS(1))                                                CRYO1     16
      WP1 = (-BZ+SQRT(BZ*BZ-4.*CZ))/2.                                   CRYO1     17
      MRTUG=(WP1+WPLB+WS(1))/(WPLB+WS(1))                                CRYO1     18
  100 ALMR=ALOG(MRTUG)                                                   CRYO1     19
      DVTUG = G*EISP(1)*ALMR/(TR+1.)                                     CRYO1     20
      RETURN                                                             CRYO1     21
      END                                                                CRYO1     22
      SUBROUTINE  PLUPD (MPLU,MRSEP,T )                                  PLUPD      2
C                                                                        PLUPD      3
C               PLUP - CARRIES SEPS PAYLOAD UP                           PLUPD      4
C                                                                        PLUPD      5
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                        /SEPVEH/   2
     * ,CHEM                                                            /SEPVEH/   3
     * ,DT                                                              /SEPVEH/   4
      INTEGER SEPS                                                       /SEPVEH/   5
      REAL MS                                                            /SEPVEH/   6
      COMMON /OUTP/ TD,TU,HCO,ICOS,MDT                                   /OUTP/     2
     * ,TLEFT(5),MPT(5),TSAVE(5),RTCAP(5),MPTSV(5)                       /OUTP/     3
      REAL MDT,MPT,MPTSV,ICOS                                            /OUTP/     4
      REAL MPLU,MRSEP,MPT1                                               PLUPD      8
      MPT1 = ((MPT(SEPS)+MS+MPLU)/MRSEP) -(MS+MPLU)                      PLUPD      9
      T = (MPT(SEPS) - MPT1)/(86400.*DT)                                 PLUPD     10
      TLEFT(SEPS) = TLEFT(SEPS) - T                                      PLUPD     11
      MPT(SEPS) = MPT1                                                   PLUPD     12
      RETURN                                                             PLUPD     13
      END                                                                PLUPD     14
      SUBROUTINE  SEPDV (HCO,ICOS,DVSEP,MRSEP )                          SEPDV      2
C                                                                        SEPDV      3
C               SEPDV - CALCULATES THE REQUIRED SEP DELTA VELOCITY       SEPDV      4
C                       NEEDED FOR SYNC EQ. AND THE CORRESPONDING        SEPDV      5
C                       MASS RATIO.                                      SEPDV      6
C                                                                        SEPDV      7
C               INPUT                                                    SEPDV      8
```

```
C                       HCO  ORBIT ALTITUDE
C                       ICOS  INCLINATION
C
C               OUTPUT
C                       DVSEP THE SEP DELTA V
C                       MRSEP THE MASS RATIO
C
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP
     * ,CHEM
     * ,DT
      INTEGER SEPS
      REAL MS
      COMMON/MISC/G
      REAL ICOS,MU
      REAL MRSEP
      DATA HS,MU,RE,DTR/19323.,1.40765388E16,3443.9308,57.295779513/
      DATA FTPNM/6076.1155/,PI02/1.570796326794/
      RCO = (HCO+RE)*FTPNM
      RS  = (HS +RE)*FTPNM
      VCO = SQRT(MU/RCO)
      VS  = SQRT(MU/RS)
      CICO = COS(PI02*ICOS/DTR)
      DVSEP = SQRT(VCO**2+VS**2-(VS+VS)*VCO*CICO)
      MRSEP = EXP(DVSEP/(G*SISP))
      RETURN
      END
      SUBROUTINE  INTORB (DVTUG,HCO,ICOS )
C
C               INTORB - AN INTERPOLATION SCHEME TO DETERMINE
C                        THE OPTIMUM CHANGEOVER ORBIT ALTITUDE
C                        AND INCLINATION.
C
C               INPUT
C                       DVTUG - TUG DELTA V
C
C               OUTPUT
C                       HCO   ALTITUDE OF CHANGEOVER ORBIT
C                       ICOS  INCLINATION OF CHANGEOVER ORBIT
C
      COMMON/TABLE/TUGDV(20)
      REAL ICOS,INC(20),ALT(20)
      DATA          TUGDV/ 10295.74,10600.0,10900.0,11200.0,11500.0,
     X                     11800.0,12100.0,12400.0,12700.0,13000.0,
     X                     13300.0,13600.0,13835.17, 7* 0.0/
      DATA          ALT   / 8000.0,8000.0,8000.0,8000.0,8000.0,8500.0,
     X                     9500.0,10500.0,11500.0,13000.0,14500.0,
     X                     17000.0,18000.0, 7*0.0/
      DATA          INC   / 28.5,19.6,15.8,12.8,10.14,8.86,8.52,7.67,
     X                     6.4,5.5, 3.87, 2.45, 8* 0.0 /
C
C               FIND THE RANGE OF DELTA V
C
      DO 20 NP1 = 2,12
      IF (DVTUG .LE. TUGDV(NP1))GO TO 30
   20 CONTINUE
C
```

SEPDV 9
SEPDV 10
SEPDV 11
SEPDV 12
SEPDV 13
SEPDV 14
SEPDV 15
/SEPVEH/ 2
/SEPVEH/ 3
/SEPVEH/ 4
/SEPVEH/ 5
/SEPVEH/ 6
SEPDV 17
SEPDV 18
SEPDV 19
SEPDV 20
SEPDV 21
SEPDV 22
SEPDV 23
SEPDV 24
SEPDV 25
SEPDV 26
SEPDV 27
SEPDV 28
SEPDV 29
SEPDV 30
INTORB 2
INTORB 3
INTORB 4
INTORB 5
INTORB 6
INTORB 7
INTORB 8
INTORB 9
INTORB 10
INTORB 11
INTORB 12
INTORB 13
INTORB 14
INTORB 15
INTORB 16
INTORB 17
INTORB 18
INTORB 19
INTORB 20
INTORB 21
INTORB 22
INTORB 23
INTORB 24
INTORB 25
INTORB 26
INTORB 27
INTORB 28
INTORB 29
INTORB 30
INTORB 31
INTORB 32

```
C                FOUND THE RANGE COMPUTE THE ALT AND INC.                    INTORB    33
                                                                             INTORB    34
   30 NPO = NP1 - 1                                                          INTORB    35
      FRAC = ( DVTUG - TUGDV(NPO ) ) / (TUGDV(NP1) - TUGDV(NPO ))            INTORB    36
      HCO  =  ALT(NPO )  + FRAC* (ALT(NP1) - ALT(NPO ))                      INTORB    37
      ICOS =  INC(NPO )  + FRAC* (INC(NP1) - INC(NPO ))                      INTORB    38
      RETURN                                                                 INTORB    39
      END                                                                    INTORB    40
      SUBROUTINE CON(I,K)                                                    CON        2
      K=0                                                                    CON        3
      IF(I.EQ.1H ) RETURN                                                    CON        4
      K=100                                                                  CON        5
      IF(I.EQ.4H    0) K=0                                                   CON        6
      IF(I.EQ.4H    1) K=1                                                   CON        7
      IF(I.EQ.4H    2) K=2                                                   CON        8
      IF(I.EQ.4H    3) K=3                                                   CON        9
      IF(I.EQ.4H    4) K=4                                                   CON       10
      IF(I.EQ.4H    5) K=5                                                   CON       11
      IF(I.EQ.4H    6) K=6                                                   CON       12
      IF(I.EQ.4H    7) K=7                                                   CON       13
      IF(I.EQ.4H    8) K=8                                                   CON       14
      IF(I.EQ.4H    9) K=9                                                   CON       15
      IF(I.EQ.4H   10) K=10                                                  CON       16
      IF(I.EQ.4H   11) K=11                                                  CON       17
      IF(I.EQ.4H   12) K=12                                                  CON       18
      IF(I.EQ.4H   13) K=13                                                  CON       19
      IF(I.EQ.4H   14) K=14                                                  CON       20
      IF(I.EQ.4H   15) K=15                                                  CON       21
      IF(I.EQ.4H   16) K=16                                                  CON       22
      IF(I.EQ.4H   17) K=17                                                  CON       23
      IF(I.EQ.4H   18) K=18                                                  CON       24
      IF(I.EQ.4H   19) K=19                                                  CON       25
      IF(I.EQ.4H   20) K=20                                                  CON       26
      IF(I.EQ.4H   21) K=21                                                  CON       27
      IF(I.EQ.4H   22) K=22                                                  CON       28
      IF(I.EQ.4H   23) K=23                                                  CON       29
      IF(I.EQ.4H   24) K=24                                                  CON       30
      IF(I.EQ.4H   25) K=25                                                  CON       31
      IF(I.EQ.4H   26) K=26                                                  CON       32
      IF(I.EQ.4H   27) K=27                                                  CON       33
      IF(I.EQ.4H   28) K=28                                                  CON       34
      IF(I.EQ.4H   29) K=29                                                  CON       35
      IF(I.EQ.4H   30) K=30                                                  CON       36
      IF(I.EQ.4H   31) K=31                                                  CON       37
      IF(I.EQ.4H   32) K=32                                                  CON       38
      IF(I.EQ.4H   33) K=33                                                  CON       39
      IF(I.EQ.4H   34) K=34                                                  CON       40
      IF(I.EQ.4H   35) K=35                                                  CON       41
      IF(I.EQ.4H   36) K=36                                                  CON       42
      IF(I.EQ.4H   37) K=37                                                  CON       43
      IF(I.EQ.4H   38) K=38                                                  CON       44
      IF(I.EQ.4H   39) K=39                                                  CON       45
      IF(I.EQ.4H   40) K=40                                                  CON       46
      IF(I.EQ.4H   41) K=41                                                  CON       47
      IF(I.EQ.4H   42) K=42                                                  CON       48
      IF(I.EQ.4H   43) K=43                                                  CON       49
      IF(I.EQ.4H   44) K=44                                                  CON       50
```

```
      IF(I.EQ.4H   45)  K=45                                          CON        51
      IF(I.EQ.4H   46)  K=46                                          CON        52
      IF(I.EQ.4H   47)  K=47                                          CON        53
      IF(I.EQ.4H   48)  K=48                                          CON        54
      IF(I.EQ.4H   49)  K=49                                          CON        55
      IF(I.EQ.4H   50)  K=50                                          CON        56
      RETURN                                                          CON        57
      END                                                             CON        58
```

13.21.45. I094415 LQ23 0004350 LINES, CU=.00030*0004350=001.3050 08/19/75 *** END OF LIST ***

AEROSPACE CORPORATION

INTERNAL DISTRIBUTION LIST

(REFERENCE: COMPANY PRACTICE 7-21-1)

REPORT TITLE
Manned Systems Utilization Analysis (Study 2.1) Final Report
Volume V: Program Listing for the LOVES Computer Code

REPORT NO.	PUBLICATION DATE	SECURITY CLASSIFICATION
ATR-76(7361)-1, Vol V, formerly ATR-74(7341)-7)	1 September 1975	

(NOTE: FOR OFF-SITE PERSONNEL, SHOW LOCATION SYMBOL, e.g. JOHN Q. PUBLIC/VAFB)

N. R. Campbell	(1)
T. J. Lang	(1)
L. T. Stricker	(1)
S. T. Wray	(3)
R. R. Wolfe	(2)

APPROVED BY Robert R Wolfe DATE 8/31/75

SHEET ____ OF ____

AEROSPACE FORM 2394 REV 7-66

THE AEROSPACE CORPORATION

EXTERNAL DISTRIBUTION LIST

(REFERENCE: COMPANY PRACTICE 7-21-1)

REPORT TITLE
Manned Systems Utilization Analysis (Study 2.1) Final Report
Volume V: Program Listing for the LOVES Computer Code

REPORT NO.	PUBLICATION DATE	SECURITY CLASSIFICATION
ATR-76(7361)-1, Vol V formerly ATR-74(7341)-7)	1 September 1975	Unclassified

MILITARY AND GOVERNMENT OFFICES	ASSOCIATE CONTRACTORS AND OTHERS

(NOTE: SHOW FULL MAILING ADDRESS; INCLUDE ZIP CODE, MILITARY OFFICE SYMBOL, AND "ATTENTION" LINE.)

NASA Scientific & Technical
Information Facility (3)
P. O. Box 33
College Park, Maryland 20740

NASA - Headquarters
Washington, D.C. 20546

Fred Roberts, Code MTS (50)

New Technology Rep., Code KT (1)

NASA
Mr. Duncan Collins
P. O. Box 92960
Worldway Postal Center
Los Angeles, CA 90000
Building 120, Rm 1406B

☐ THIS DISTRIBUTION LIST PERMITS ONLY PRIMARY DISTRIBUTION OF THIS TOR.

AFR 80-45 DISTRIBUTION STATEMENT X'D BELOW APPLIES

☐ NO DISTRIBUTION STATEMENT
(Classified documents only)

☐ A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

☐ B. DISTRIBUTION LIMITED TO U. S. GOV'T AGENCIES ONLY;
_____ (Reason)
_____ (Date statement applied). OTHER REQUESTS FOR THIS DOCUMENT MUST BE REFERRED TO _____ (Controlling DOD office)

APPROVED BY Robert R Wolfe (FOR THE AEROSPACE CORPORATION) DATE 8/31/75

APPROVED BY _____ (FOR COGNIZANT AF OFFICE) _____ (SYMBOL) DATE _____

IF LIST COMPRISES TWO OR MORE SHEETS, COMPLETE THIS SIGNATURE BLOCK ON LAST SHEET ONLY

SHEET _____ OF _____

AEROSPACE FORM 2380 REV 7-75